Statistical mechanics approaches to optimization and inference

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Physics (29^{th}cycle)

Statistical Mechanics approaches to optimization and inference

By

Anna Paola Muntoni

******

Supervisor(s):
Prof. Alfredo Braunstein

Doctoral Examination Committee:
Prof. M. Weigt, Referee, CNRS/UPMC, France
Prof. F. Ricci Tersenghi, Referee, Università "La Sapienza", Italy

Politecnico di Torino
2017

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Anna Paola Muntoni
2017

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgements

I would like to sincerely acknowledge my supervisor Alfredo Braunstein for offering me the opportunity of working with him, and for always being a patient and motivated teacher. His precious knowledge has been of tremendous help for my research, and for my academic and personal growth. Besides my supervisor, I would also like to thank Andrea Pagnani, whom I truly appreciated working with. I thank all my colleagues from the CMP group of Politecnico for non-trivial and ever-interesting conversations and for having shared with me all the difficulties and the achievements of these three years.

I thank my family, Stefano, Roberta, Daniele, Chiara, Mario and all my friends from Oschiri, for always being encouraging and for all the nice moments spent together during these years. Finally, I warmly thank Giulio for the infinite patience and for always believing in me; his constant support and precious advice have had a key role in this work.

Abstract

Nowadays, typical methodologies employed in statistical physics are successfully applied to a huge set of problems arising from different research fields. In this thesis I will propose several statistical mechanics based models able to deal with two types of problems: optimization and inference problems. The intrinsic difficulty that characterizes both problems is that, due to the hard combinatorial nature of optimization and inference, finding exact solutions would require hard and impractical computations. In fact, the time needed to perform these calculations, in almost all cases, scales exponentially with respect to relevant parameters of the system and thus cannot be accomplished in practice.

As combinatorial optimization addresses the problem of finding a fair configuration of variables able to minimize/maximize an objective function, inference seeks *a posteriori* the most fair assignment of a set of variables given a partial knowledge of the system. These two problems can be re-phrased in a statistical mechanics framework where elementary components of a physical system interact according to the constraints of the original problem. The information at our disposal can be encoded in the Boltzmann distribution of the new variables which, if properly investigated, can provide the solutions to the original problems. As a consequence, the methodologies originally adopted in statistical mechanics to study and, eventually, approximate the Boltzmann distribution can be fruitfully applied for solving inference and optimization problems.

The structure of the thesis follows the path covered during the three years of my Ph.D. At first, I will propose a set of combinatorial optimization problems on graphs, the Prize collecting and the Packing of Steiner trees problems. The tools used to face these hard problems rely on the zero-temperature implementation of the Belief Propagation algorithm, called Max Sum algorithm. The second set of problems proposed in this thesis falls under the name of linear estimation problems. One of them, the compressed sensing problem, will guide us in the modelling of these problems within a Bayesian framework along with the introduction of a powerful algorithm known as Expectation Propagation or Expectation Consistent in statistical physics. I will propose a similar approach to other challenging problems: the inference of metabolic fluxes, the inverse problem of the electro-encephalography and the reconstruction of tomographic images.

Contents

Part I

Statistical physics for combinatorial problems

Chapter 1

Introduction

In this chapter we will introduce the main features of statistical mechanics along with a brief review about optimization and inference problems. The link among the three disciplines is underlined by several examples in which we express both combinatorial optimization and inference problems using a language borrowed from statistical physics.

1.1 Basic notions of statistical mechanics

Statistical mechanics was born in the end of the 19^{th} century to rigorously explain the laws of thermodynamics in a probabilistic framework. Starting from a microscopical description of a system of a large number of elementary units (for instance, particles in a gas), it recovers a macroscopic viewpoint that, at equilibrium, is governed by the state laws of thermodynamics [1, 2].

Let us define the microscopic state of element i as x_i and the *space of configurations* as X such that $x_i \in X$; it fully describes the microscopic details, i.e. all the degrees of freedom, of element i. From now on we consider x_i as a discrete variable but the generalization to continuous space is straightforward. For a system of N units, the configuration space of the state $\boldsymbol{x} = \{x_i\}_{i=1,\dots,N}$ will be the product space $X_N = X \times \ldots \times X$, and thus $\boldsymbol{x} \in X_N$.

The energy function, or Hamiltonian, $H(\boldsymbol{x})$ depends on the kind of interaction among the elementary components. In the most general case the energy can be any function of the configuration that depends on many-bodies terms, we say $k-$bodies, as

$$H(\boldsymbol{x}) = \sum_{i_1,\dots,i_k} H(x_{i_1}, \dots, x_{i_k}) \tag{1.1.1}$$

where k can take any value from 1 to N. For non-interacting systems, the energy function is the sum of single elements energy, for instance

$$H(\boldsymbol{x}) = \sum_{i=1}^{N} H(x_i) \tag{1.1.2}$$

Notice that the expressions in (1.1.1) and (1.1.2) can be further simplified if one replaces the exact expression with the Taylor expansion of $H(\boldsymbol{x})$ with respect to $\boldsymbol{x}$. Once the space of configurations and the energy function are defined, we can state that, in a canonical ensemble picture, the probability that the system assumes a configuration $\boldsymbol{x}$ is given by the Boltzmann distribution

$$P(\boldsymbol{x}) = \frac{1}{Z_\beta} e^{-\beta H(\boldsymbol{x})} \tag{1.1.3}$$

where Z_β is the partition function defined as

$$Z_\beta = \sum_{\boldsymbol{x} \in X_N} e^{-\beta H(\boldsymbol{x})} \tag{1.1.4}$$

The parameter β appearing in (1.1.3) and (1.1.4) is the so-called inverse temperature defined as $\beta = \frac{1}{k_B T}$ where T is the temperature, k_B is the Boltzmann constant that, for sake of simplicity, will be omitted for the rest of the work.

The sum in (1.1.4) runs over all the possible configurations of the system and, depending of the functional form of the energy, it may lead to an *intractable* or *unfeasible* computation. This means that, although the partition function is exactly computable, the time needed to perform this calculation grows exponentially with the size N of the system. Very few statistical models can be "solved" in the sense that the partition function can be computed in polynomial time. The distinction among tractable and intractable problems is outlined in section 1.2.1.

Let us outline how a system described by the distribution in (1.1.3) behaves in the following limiting cases:

- **High temperature regime**. Performing the $\beta \to 0$ limit, one finds:

$$\lim_{\beta \to 0} P(\boldsymbol{x}) = \frac{1}{|X|} \tag{1.1.5}$$

  The Boltzmann distribution in (1.1.3) becomes a flat distribution: as a consequence all configurations are equally likely to occur.

- **Low temperature regime** In the $\beta \to \infty$ limit the Boltzmann distribution concentrates over the maximum(a) of (1.1.3), or in other words, those configurations that minimize the energy function, the ground states, dominate the Boltzmann distribution. Assuming that H has a unique minimum and defining as $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} H(\boldsymbol{x})$ the ground state, the Boltzmann distribution reads

$$\lim_{\beta \to +\infty} P(\boldsymbol{x}) = \mathbb{I}[\boldsymbol{x} = \boldsymbol{x}^*] \tag{1.1.6}$$

  where $\mathbb{I}[\cdot]$ is the indicator function that takes value 1 is its argument is true or 0 otherwise.

The behavior of the system at microscopic level and, therefore, the main aspects of the Boltzmann distributions are caught by some thermodynamic potentials whose "macroscopic" behavior reflects the microscopic one. The most important one is the (Helmholtz) free energy

$$F = -\frac{1}{\beta} \ln Z_\beta \tag{1.1.7}$$

since, as discussed in section §3.2, a system at thermal equilibrium will assume a configuration of variables such that the free energy potential is minimized. From the free energy one can derive two more thermodynamic potentials, the internal energy U and the entropy S

$$U = \frac{\partial}{\partial \beta}[\beta F] \qquad\qquad S = \beta^2 \frac{\partial}{\partial \beta} F \tag{1.1.8}$$

that are all related through the expression

$$F = U - TS \tag{1.1.9}$$

Thus

$$U = \sum_{\boldsymbol{x}} H(\boldsymbol{x}) P(\boldsymbol{x}) \qquad\qquad S = -\sum_{\boldsymbol{x}} P(\boldsymbol{x}) \ln P(\boldsymbol{x}) \tag{1.1.10}$$

where this latter expression of S is the Shannon entropy coming from information theory [3].

1.1.1 Phase transitions

The main purpose of statistical mechanics is to quantitatively study the behavior of a system in the thermodynamic limit, that is when the number of the elementary units N is very large, or, more formally, when $N \to +\infty$. Since the thermodynamic potentials

defined in section §1.1 proportionally growth with respect to the system size, it is useful to introduce an intensive measure of these potentials when $N \to +\infty$. For instance, let us define the intensive free energy functional as

$$f = \lim_{N \to +\infty} \frac{F}{N} \tag{1.1.11}$$

For finite N the partition function and the Helmholtz free energy defined in (1.1.4)(1.1.7) are analytic functions of the temperature (or of the parameter β) but in the thermodynamic limit analyticity may be not preserved. In these cases, there exists a the critical value of the (inverse) temperature β_c on which the intensive free energy is not analytic. Macroscopically, the system undergoes a phase transition and it dramatically changes its state. We can differentiate two cases:

- **First order phase transition**. The free energy is a continuous function of β but its derivative has a discontinuity in β_c;
- **Second order phase transition**. The free energy and its first derivative are continuous but the second derivative has a singularity in β_c.

In this thesis we will encounter some phase transitions. When we will investigate the solution space of combinatorial optimization and inference problems, we will face sort of phase transitions in the sense that these problems can change their "state" from solvable to unsolvable in particular settings. This is what we will show in section 2.2.1.3 for the compressed sensing problem introduced in section 2.2.1.1.

Example 1. Ising model

The Ising model is the most famous model of statistical physics. As a matter of example let us consider a discrete model of N spins subjected to pairwise interactions and an external field. The state σ_i of spin i is defined in the space of possible spin magnetic moments, $\sigma_i \in \{-1, 1\}$. The Hamiltonian is defined as

$$H(\boldsymbol{\sigma}) = -\sum_{(i,j)} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i \tag{1.1.12}$$

where the sum over (i, j) runs over all coupled spins. The parameters J_{ij} govern the strength of the coupling and they take positive (negative) value for ferromagnetic (paramagnetic) interaction. With h_i we denote the external field applied to spin i. The partition function associated with this Hamiltonian is given by

$$Z_\beta = \sum_{\boldsymbol{\sigma}} e^{-\beta H(\boldsymbol{\sigma})} \tag{1.1.13}$$

that is not tractable in general cases.

1.2 Optimization problems

1.2.1 NP versus P

The link between computer science and statistical mechanics has recently produced a breeding ground for the study of combinatorial problems [4]. Since the beginning of the Sixties, algorithms complexity played a central role in the understanding of the performance of digital calculus, addressing the question of how the time and space consumption is linked to the input size of the problem.

This study has produced the important distinction between *polynomial* time (P) problems and *non-deterministic polynomial* (NP) time problems. By definition, for the first class there exist algorithms that guarantee a solution for any instance of the problem in a *feasible* time, i.e. that scales polynomially with the input size; for the second class we can only verify that a candidate solution is indeed a solution. We say that some instances of an NP problem are *intractable* as the resolution process may need a computing time that can even exponentially grow with respect to the input size. Going deeper in the classification we can briefly introduce the classes of NP-complete and NP-hard problems. A problem belongs to the NP-hard class if any problem in NP can be converted to it in feasible time. An NP-complete problem is an NP-hard problem and it is itself NP, meaning that checking if a trial solution is a solution requires a time that scales polynomially. A consequence of this is that the existence of a polynomial algorithm for any NP-complete problem implies $NP = P$. So far, no such an algorithm for any NP-complete problem has been found and, although no proof exists, the hypothesis that $NP \neq P$ seems hard to refute. Another consequence is that there exist "harder" problems belonging to the NP-hard class, but not in NP, such that even checking a solution requires unfeasible time.

Cook's paper [5] proved that the SAT-problem (the decision problem of determining whether a Boolean logic formula can be satisfied or not) belongs to the NP-complete class; to prove that any other NP problem is NP-complete, it suffices to prove that satisfiability can be reduced to it. That's what Karp did in 1972 [6] where 21 combinatorial problems have been classified in the NP-complete class (and since then, many more). Several combinatorial decision problems belong to the NP-complete class but its optimization version falls outside of NP. To underline this distinction we show in example 2 the decision and the optimization version of the traveling salesman problem.

Example 2. Traveling salesman problem

Consider a salesman that wants to visit N cities in the following way: starting from the first one, he would like to visit any other city once and ends the tour with the first one. With each possible tour we associate a cost that is the sum of the kilometers covered by the salesman. We can formulate two versions of the problem:

- **Optimization problem**: What is the path of minimum cost? There is not a tractable strategy to verify that a candidate solution is the optimum: in order to identify the minimum circuit satisfying the constraints one needs to explore all the paths. This is also unfeasible. In fact, a brute force search algorithm will explore all the possible closed path in $\mathcal{O}(N!)$ steps. Performances of more sophisticated algorithms will require $\mathcal{O}\left(N^2 2^N\right)$ iterations [7]. This version of the problem is NP-hard.

- **Decision problem**: Given a cost L, is there a tour with cost smaller than L? To verify that a trial closed path satisfies the hard constraint of visiting once each city and compare the costs are surely feasible. Looking for a solution is instead intractable as before. The decision problem is thus NP-complete.

1.2.2 Link to statistical mechanics

The classification presented in section 1.2.1 deals with universal properties: when we say that "this problem needs a computing time which is exponential" means essentially that there exists at least one difficult instance of this kind; it is a worst case analysis. Fortunately, the occurrence of difficult or easy instances really depends on which problem we are dealing with so we can wonder what happens in typical cases: how frequent is an easy instance? Can I use some methods to solve this subset of instances? This is where statistical mechanics enters into the picture. In particular, the first link between combinatorial optimization problems and statistical physics has been made in [8] and typically it is stated as follows. A combinatorial optimization problem consists in finding a particular state or configuration that minimizes an objective function. In a statistical physics picture it corresponds to finding the ground state of a system whose energy is the objective function of the combinatorial problem. More formally, we aim at determining the configuration $\boldsymbol{x}^*$ that minimizes the objective function $H(\boldsymbol{x})$, or equivalently, dominates the Boltzmann distribution $P_\beta(\boldsymbol{x}) = \frac{1}{Z_\beta} e^{-\beta H(\boldsymbol{x})}$ for $\beta \to +\infty$ as we have explained in section §1.1.

This is key observation that will be highly exploited in the problems faced in this thesis. To give an example, we describe in example 3 a way of treating the graph

coloring problem in a statistical mechanics framework; this needs to be understood as an introduction to the mapping described in section §4.2 and section §5.2.

Example 3. Graph coloring

Given a graph $\mathcal{G}(V, E)$ and a set of Q colors, the problem consists in assigning to any node $i \in V$ a color such that its neighbors $j \in \partial i$ have different colors. For general graphs, determining whether a "colorable" assignment exists is one of the 21 NP-complete problems listed by Karp in [6].

Let us define $x_i \in Q$ the color of node i and $\boldsymbol{x} = \{x_i\}_{i=1,\dots,N}$ a possible configuration of the N nodes; the "Hamiltonian" or cost function that counts how many constraints are not satisfied is given by

$$H(\boldsymbol{x}) = \sum_{(i,j)\in E} \delta_{x_i,x_j} \tag{1.2.1}$$

Thus the Boltzmann distribution associated with the variables $\boldsymbol{x}$ is

$$P(\boldsymbol{x}) = \frac{1}{Z_\beta} e^{-\beta \sum_{(i,j)} \delta_{x_i,x_j}} \tag{1.2.2}$$

The ground state will be composed by those configurations, in any, of zero-energies; other configurations, with positive energies, will not exactly solve the problem. Notice that if we perform the $\beta \to +\infty$ limit, we force the system to lie on the ground state.

The modeling is certainly not unique. For instance, in order to stress the hard constraint imposed to each couple of neighbors $(i, j) \in E$, we can eventually define a joint probability distribution of the variables as

$$P(\boldsymbol{x}) = \frac{1}{Z} \prod_{(i,j)\in E} \left(1 - \delta_{x_i,x_j}\right) \tag{1.2.3}$$

1.3 Statistical inference

Statistical inference is the process of deducing properties of an underlying distribution by analysis of data. Very often the information at our disposal is affected by uncertainty or it partially describes the properties that we would like to infer; as a consequence, we need to investigate the problem in a probabilistic framework [9, 10]. To give an example, suppose of having a coin and a series of "head" or "tail" outcomes; how can we determine if the coin is fair? Although there is not a process that can deterministically answer "yes" or "no" to this question, we can investigate what is most probable value for the probability of "heads" or "tails" given a series of outcomes and thus decide if the coin is biased or not.

Despite the fact that our knowledge suffers from uncertainties about all the details of the problem in interest, we assume that the process can be well explained by an hypotheses or a model. Among all the approaches to statistical inference the more intuitive is the so-called Bayesian inference; its name clearly derives from the use of Bayes' theorem of probability theory. Not only this method provides a powerful connection between observations and unknowns, but it can, in principle, estimates the goodness of our hypothesis, or model, that, we think, describes the problem.

Suppose of having access to M measurements $\{y_i\}_{i=1,\dots,M} = \boldsymbol{y}$ generated by a system that we assume is well described by hypothesis $\mathcal{H}$. We denote as $\{\tilde{x}_i\}_{i=1,\dots,N} = \tilde{\boldsymbol{x}}$ the continuous or discrete variables that we aim at inferring. Let us define $\boldsymbol{x}$ our estimate of $\tilde{\boldsymbol{x}}$. We ask ourselves, what is the probability distribution of the unknown variables $\boldsymbol{x}$ given that we have observed $\boldsymbol{y}$ and our model $\mathcal{H}$ is correct? Bayes' theorem states that

$$P(\boldsymbol{x}|\boldsymbol{y},\mathcal{H}) = \frac{P(\boldsymbol{y}|\boldsymbol{x},\mathcal{H})\,P(\boldsymbol{x}|\mathcal{H})}{P(\boldsymbol{y}|\mathcal{H})} \tag{1.3.1}$$

Probabilities entering in (1.3.1) have special names in the context of statistical inference:

- $P(\boldsymbol{x}|\boldsymbol{y},\mathcal{H})$ is the *posterior* probability;
- $P(\boldsymbol{y}|\boldsymbol{x},\mathcal{H})$ is the *likelihood*. For fixed $\boldsymbol{x}$ it is a probability over $\boldsymbol{y}$, otherwise it is a function of both;
- $P(\boldsymbol{x}|\mathcal{H})$ is the *prior* probability. It includes the prior knowledge, in any, of the unknowns. Otherwise it is taken as a uniform distribution;
- $P(\boldsymbol{y}|\mathcal{H})$ is called *evidence*. In the context of model selection, i.e. estimate how good is our model, it plays a main role.

Our model $\mathcal{H}$ is formally characterized by a set of parameters, let us define them as $\boldsymbol{\theta}$, which must be carefully determined in order to the model to well describe the problem in interest. For most of the problems and applications presented in this work, we will drop the "$\mathcal{H}$" on the probabilities except when we will try to infer the parameters from the data.

1.3.1 A statistical physics picture

Historically, the first analogy between statistical inference and statistical physics can be found in the seminal work of Shannon [3] where the term "entropy" has been used to

quantify the amount of information in an inference problem. More importantly, Jaynes in [11] argued that, according to maximum entropy principle, statistical mechanics has not to be thought as a physical theory depending on its assumptions but it is a special case of a general Bayesian inference theory [9]. Nowadays, the two disciplines are more and more connected as problems arising from one field can be treated using methodologies of the other [1, 12].

Mathematically, a straight-forward link between statistical inference and statistical physics can be established re-phrasing the posterior distribution (to the power of β) in (1.3.1) as a Boltzmann distribution of the type

$$P\left(\boldsymbol{x}|\boldsymbol{y}\right)^{\beta} = \frac{1}{Z\left(\boldsymbol{y}\right)} e^{\beta \log P(\boldsymbol{y}|\boldsymbol{x}) + \beta \log P(\boldsymbol{x})} = \frac{1}{Z\left(\boldsymbol{y}\right)} e^{-\beta H(\boldsymbol{x},\boldsymbol{y})} \tag{1.3.2}$$

where $Z\left(\boldsymbol{y}\right)$ is the partition function and $H\left(\boldsymbol{x},\boldsymbol{y}\right)$ is the Hamiltonian of a system of interacting units that, in this case, are both $\boldsymbol{x}$ and $\boldsymbol{y}$. The parameter β is eventually set to $+\infty$ when we want to investigate the most probable configurations $\boldsymbol{x}^* = \arg\max P\left(\boldsymbol{x}|\boldsymbol{y}\right)$ that correspond to the ground states of the system with Hamiltonian $H\left(\boldsymbol{x},\boldsymbol{y}\right)$. This model is very general as can encompass both discrete and continuous variables. We will use this kind of mapping in section 2.2.1.1, section §7.1, section §8.1 and section §9.1.

Chapter 2

Problems

In this chapter we will provide a formal description of the problems faced in this thesis together with a brief introduction to their applications.

2.1 Steiner tree problems

The minimum Steiner Tree problem (MStP) is an important combinatorial optimization problem. It consists in finding a connected sub-graph, within a weighted graph, able to connect a given subset of nodes, called terminal nodes, at minimum cost. With a solution of this problem we associate an objective function that represents the cost of connection and it is given by the sum of weights associated with the edges of the sub-graph. The decision problem of determining whether there exists a solution of cost less then a given threshold is one of the 21 NP-complete problems identified by Karp in [6]. The optimization problem of finding the connected sub-graph of minimum cost is instead NP-hard. Assuming that we want to minimize the objective function with the constraint of connecting all terminals, it is easy to show that the solution is an acyclic connected sub-graph, that is a tree. In fact, starting from a loopy sub-graph in which all terminals are connected we can prune certain edges, lowering the energy and keeping the connection.

There exist many different variants of the MStP depending on the additional constraints that nodes, edges or even the degree of the tree must satisfy. In this work we deal with a generalization of the MStP that is the Prize-Collecting Steiner tree problem (PCStP) where, instead of dealing with terminals, we are given a graph with "prizes" associated with nodes. The objective function is now lowered each time we add a "prized" node: the optimal configuration will be given by the best trade-off between the cost of the solution and the reward gained by the insertion of nodes.

Both MStP and PCStP have been recently used in several different frameworks to model a huge set of problems. Consider, for instance, the problem of optimally distributing heating or fiber optics in a city [13]. Customers (or potential customers) can be seen as prized nodes in a graph, the street network of the town. Edges will be weighted according to the cost of the connection of pipes and cables. A solution of the MStP on this type of graph provides an optimal set of potential customers along with the most efficient architecture of the pipes or cables. In addition to the optimal power distribution problem, we mention the research of protein associations in cell signaling [14, 15] and the circuit design problem [16].

The most popular approaches to the PCStP are the use of linear programming [17] and optimized heuristics [18]. We present in 3.6.1.1.1 the standard Goemans-Williamson heuristics [19] as a background to the Max-Sum modified heuristics presented in section §4.4. A cavity method has been used in [20, 21] to study the statistical properties and performances of an implementation of the Max sum algorithm in section 3.3.4 for this problem. The approach mainly relies on "pointer variables" associated with nodes of the graph.

Another variant we are going to present is the Packing of Steiner trees where we aim at finding multiple trees within the graph. These trees "interact" in the sense that nodes and/or edges belonging to a certain solution must satisfy additional constraints. In particular, in the Vertex-disjoint Steiner tree problem (V-StP) once a node is part of one solution it cannot be shared by other trees. As a consequence, also edges can belong to at most one tree. Differently, in the Edge-disjoint Steiner tree problem (E-StP) edges of a solution-tree cannot be used by other trees but nodes can be shared by different trees.

In addition to its mathematical interest, a lot of attention is devoted to the Packing of Steiner trees since several layout design issues arising from Very Large Systems Integrated (VLSI) circuits can be mapped into variants of this combinatorial optimization problem [16]. Graphs underling these circuits are typically 2D or 3D grid graphs where the terminals assignment usually satisfies some working conditions.

2.1.1 The Prize-Collecting Steiner tree problem

Let us give a formal definition of the PCStP. Given a graph $\mathcal{G}(V, E)$ with prizes $c_i \geq 0$ associated with each node $i \in V$ and weights $w_{ij} > 0$ on each edge $(i, j) \in E$, we address the problem of finding a tree $\mathcal{T}(V_T, E_T)$, where $V_T \subseteq V$, $E_T \subseteq E$ and the objective function

$$H(V_T, E_T) = \sum_{(i,j)\in E_T} w_{ij} - \sum_{i\in V_T} c_i \tag{2.1.1}$$

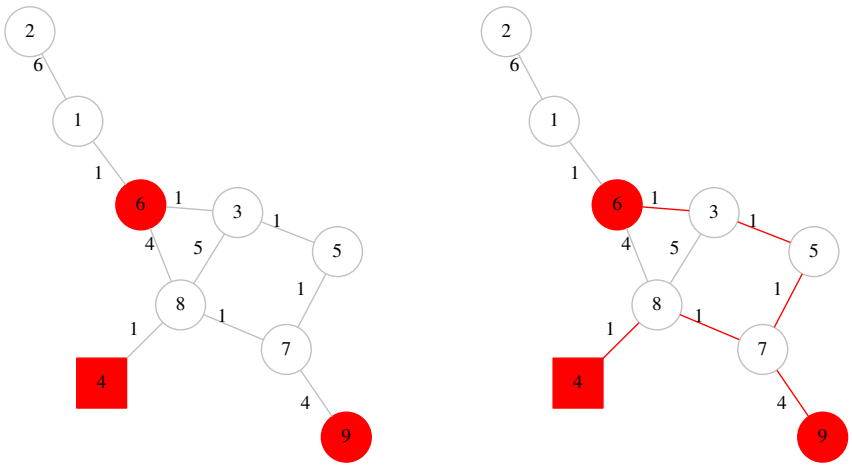


Figure 2.1.1: Instance of the MStP on the left and the optimal solution on the right

is minimized. Thus, we seek

$$\mathcal{T}^* = \arg \min_{(V_T, E_T)} H\left(V_T, E_T\right) \tag{2.1.2}$$

Notice that if we impose very large prizes to the nodes, for instance for $c_i \to +\infty$, we recover the MStP. Nodes having $c_i > 0$ are often called profitable nodes.

In chapter 4 we present an algorithm, based on the Max Sum algorithm presented in section 3.3.4, able to efficiently find a "bounded" and "rooted" solution. We mean that among all the profitable nodes, one is selected to be the *root* node of the tree and this needs to be surely included in the solution. We can also define the diameter of the tree as the maximum distance between the root and any other node within the tree. With "bounded" we indicate that the diameter of the solution can be at most equal to a constant; if this constant is D, we say that the tree is D-bounded.

As a matter of example, we propose in figure 2.1.1 (left plot) a small graph in which we solve the MStP. Red nodes are the terminals (node "4" is the root) while the numbers close to the edges are the weights of the graph. The solution is plotted on the right where the edges of the solution are now red. Nodes different from the terminals that are members of the solution are called Steiner nodes; in this example nodes $\{3, 5, 7, 8\}$ are Steiner nodes.

2.1.2 Packing of Steiner trees problems

In the following we will give a formal definition of the packing of Steiner Trees problems.

Given a weighted graph $\mathcal{G}(V, E, M)$ containing M subsets of vertices with non-negative real prizes $\{c_i^\mu : i \in V,\ \mu = 1, \ldots, M\}$ and real positive weights $\left\{w_{ij}^\mu : (i,j) \in E,\ \mu = 1, \ldots, M\right\}$ on edges, we consider the problem of finding M connected sub-graphs $\mathcal{G}_\mu = (V_\mu, E_\mu)$ that minimize the following cost or energy function:

$$H = \sum_\mu \left[\sum_{i \in V \setminus V_\mu} c_i^\mu + \sum_{(i,j) \in E_\mu} w_{ij}^\mu \right] \tag{2.1.3}$$

This definition of the problem is as general as possible since node prizes and edge costs can depend on sub-graph μ and, for directed graph, we can admit $w_{ij}^\mu \neq w_{ji}^\mu$. In the following we interpret the solution-trees as networks that allow terminals to "communicate" thus each sub-graph $\mathcal{G}_\mu$ is a "communication" μ flowing within the graph.

Additionally, subsets $\mathcal{G}_\mu$ must satisfy some interaction constraints. In the *Vertex-disjoint Steiner Trees Problem* (V-DStP), vertex-sets V_μ must be pairwise disjoint, i.e. $V_\mu \cap V_\nu = \emptyset$ if $\mu \neq \nu$ and, consequently, also edge sets will be pairwise disjoint whereas in the *Edge-disjoint Steiner Trees Problem* (E-DStP), only edge sets must be pairwise disjoints, i.e. $E_\mu \cap E_\nu = \emptyset$ if $\mu \neq \nu$, but vertex sets can overlap. In chapter 5 we will present three algorithms to solve the V-DStP and the E-DStP based on the zero-temperature implementation of the Belief Propagation, the Max Sum algorithm outlined in section 3.3.4.

2.2 Linear estimation problems

The problems that will be described in the following part of the chapter can be seen as particular cases of a more general problem, the *linear estimation problem* (LEP). This type of problem consists in determining a vector $\boldsymbol{x} \in \mathbb{R}^N$ that satisfies a system of linear equations of the type

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y} \tag{2.2.1}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\boldsymbol{y} \in \mathbb{R}^M$ are known and $N > M$. The system in (2.2.1) is ill-posed and, mathematically speaking, there exists an infinite number of solutions satisfying the linear constraints; nonetheless the domain of research can be significantly reduced if one makes further assumptions concerning the desired solution.

These problems can be faced using very different techniques. We will shortly show how to deal with LEP using linear programming in section §3.6.1 but the main approaches

utilized in this thesis rely on the Bayesian inference. We will encode within the posterior probability of the unknowns all the available information about the system along with the hard constraints that variables $\boldsymbol{x}$ must satisfy.

In the following we will formulate several inverse problem as particular cases of the linear estimation problem. Among all, the compressed sensing problem will be mapped here into a Bayesian inference problem and in section 2.2.1.2 we will show some known results of this particular problem to be thought as an introductory part of the preliminary results shown in chapter 6.

2.2.1 Compressed sensing

In the framework of signal acquisition, the compressed sensing (CS) problem [22] addresses the question of determining what is the best procedure to sample, compress and save the data without losing information. Applications of the CS problem arise from very different fields: from image processing [23, 22] to astronomy [24] and biology [25].

The problem setting is typically the following. A receiver acquires some measurements $\boldsymbol{y} = \{y_m\}_{m=1,\dots,M}$ emitted by a noisy device. Typically, we know how the machine works: starting from a signal $\boldsymbol{s} = \{s_n\}_{n=1,\dots,N}$, it makes linear operations that can be formally described as multiplication by an $M \times N$ matrix $\mathbf{F}$ (if this matrix is not given, there exist several procedures able to design one [26]). Since the output of the operation is affected by some additive noise $\boldsymbol{\varepsilon}$, each component of the measurements vector is mathematically described as

$$y_m = \sum_{n=1}^{N} F_{mn} s_n + \varepsilon_m \quad m = 1, \dots, M \tag{2.2.2}$$

In practical cases, the system of equations expressed in (2.2.2) is ill-posed, being $N > M$. In the compressed sensing framework we assume that the original signal is K-*sparse*, meaning that only a fraction $\rho = \frac{K}{N}$ of elements of $\boldsymbol{s}$ is non-zeros.

We will present how to cope the CS problem in a Bayesian framework in section 2.2.1.1; approaches for solving this problem are illustrated in section 3.4.1, section 3.6.1.2 and chapter 6.

2.2.1.1 Bayesian approach to CS problem

Let us call $\boldsymbol{x}$ our prediction about the signal $\boldsymbol{s}$ in (2.2.2) and let us specify the formal expression of the posterior $P(\boldsymbol{x}|\boldsymbol{y})$ in (1.3.1). The likelihood function remarks the linear constraint;in the noisy case, we can express the probability of observing $\boldsymbol{y}$ given $\boldsymbol{x}$ as the probability of observing the noise. Thus if we assume that each component ε_m is

distributed according to a Gaussian distribution of zero mean and variance Δ, we obtain

$$P(\boldsymbol{y}|\boldsymbol{x}) \propto \prod_m \frac{1}{\sqrt{2\pi\Delta}} e^{-\frac{1}{2\Delta}\left(y_m - \sum_n F_{mn} x_n\right)^2} \tag{2.2.3}$$

The noiseless case can be encoded taking the limit of $\Delta \to 0$ of equation (2.2.3) that corresponds to exactly impose the linear constraint $\boldsymbol{y} = \mathbf{F}\boldsymbol{x}$. The prior $P(\boldsymbol{x})$ enforces the sparsity of the solution; if we define $\rho = \frac{K}{N}$ the fraction of non-zeros components and we assume independence among them, we obtain

$$P(\boldsymbol{x}) \propto \prod_n \left[(1-\rho)\,\delta(x_n) + \rho\varphi(x_n)\right] \tag{2.2.4}$$

where $\delta(\cdot)$ is the Dirac delta function and $\varphi(\cdot)$ is a finite variance function to be determined. This type of prior is often called L_0 regularization. If we assume that non-zeros entries are smooth and distributed according to a Gaussian distribution with zero mean and variance λ we can write the posterior as:

$$P(\boldsymbol{x}|\boldsymbol{y}) \propto e^{-\frac{1}{2\Delta}(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})^T(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})} \prod_n \left[(1-\rho)\,\delta(x_n) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{x_n^2}{2\lambda}}\right] \tag{2.2.5}$$

2.2.1.2 Different priors

The L_0 prior introduced in (2.2.4) "directly" imposes the sparsity constraint, in the sense that the sought solution have exactly ρN non-zeros components. Let us write the posterior in equation (2.2.5) as in the statistical mechanics picture outlined in section 1.3.1

$$P(\boldsymbol{x}|\boldsymbol{y}) \propto e^{-\beta H(\boldsymbol{x},\boldsymbol{y})} \tag{2.2.6}$$

where

$$H(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2\Delta}(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})^T(\boldsymbol{y}-\mathbf{F}\boldsymbol{x}) - \sum_n \ln\left[(1-\rho)\,\delta(x_n) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{x_n^2}{2\lambda}}\right] \tag{2.2.7}$$

Due to the presence of the Dirac delta distribution, this energy function is not easy to treat with standard minimization approaches. For this reason one can "relax" the sparsity constraint imposing a different kind of prior to the $\boldsymbol{x}$ variables which corresponds to replacing the logarithm term in the Hamiltonian with a convex function of $\boldsymbol{x}$. We will show here two formulations:

- L_1 **regularization.** This prior is also known as Laplace or LASSO (least absolute shrinkage and selection operator) [27] prior and consists in the following distribution

$$P_{L_1}(\boldsymbol{x}) \propto \prod_n e^{-\beta_1 |x_n|} = e^{-\beta_1 \|\boldsymbol{x}\|_1} \tag{2.2.8}$$

where the function $\|\boldsymbol{x}\|_1$ is the L_1-norm. The parameter $\beta_1 > 0$ governs the sparseness of the solution.

- L_2 **regularization.** Here we replace the $\|\boldsymbol{x}\|_1$ in (2.2.8) with the $\|\boldsymbol{x}\|_2$ as

$$P_{L_2}(\boldsymbol{x}) \propto \prod_n e^{-\beta_2 x_n^2} = e^{-\beta_2 \|\boldsymbol{x}\|_2} \tag{2.2.9}$$

The posterior is this case assume the functional form of a multivariate Gaussian distribution.

These formulations of the prior allow to treat the CS problem with Linear Programming techniques as shown in section 3.6.1.2.

2.2.1.3 Thermodynamic limit

Both using mathematical techniques [28] and replica symmetric analysis [29, 30] it is possible to study what happens to the space of solutions of the CS problem in the thermodynamic limit, that is for $N \to +\infty$ and $M \to +\infty$ and fixed measurement rate $\alpha = \frac{M}{N}$.

Let us suppose that the matrix of measurement $\mathbf{F}$ is composed by independently distributed Gaussian random variables and let us work in the noiseless regime: the system in (2.2.2) describes a set of linear independent equations. If $M \geq K$, we can try to solve the system $\boldsymbol{y} = \mathbf{F}\boldsymbol{x}$ for any of the possible $\binom{N}{K}$ assignments of the non-zero entries of $\boldsymbol{x}$ and determine which $\boldsymbol{x}$ satisfies the linear constraint. However, this exhaustive enumeration approach is impractical as the operations needed to seek a solution scales exponentially with N. Notice also that for $M < K$, that is when the measurement rate is smaller then the fraction of non-zero of the signal, it is impossible to solve the problem. Thus there exists a limit of perfect reconstruction, $\alpha = \rho$, beyond which, that it for $\alpha \geq \rho$, it is, in principle, possible to retrieve the signal. This is what we are seeking when we impose an L_0 prior on the variables $\boldsymbol{x}$. What happens when the prior is instead L_1 or L_2? We plot in figure 2.2.1 the phase diagram of the space of solution of this problem as explained in [30]. Blue, red, and green lines correspond to the critical values of α as a function of the sparsity ρ when we use the L_0, the L_1 and L_2 regularizations respectively. We notice that for the L_1 prior the portion of the space between the red and the blue line is not accessible even though we are beyond the theoretical limit of the L_0 line. This

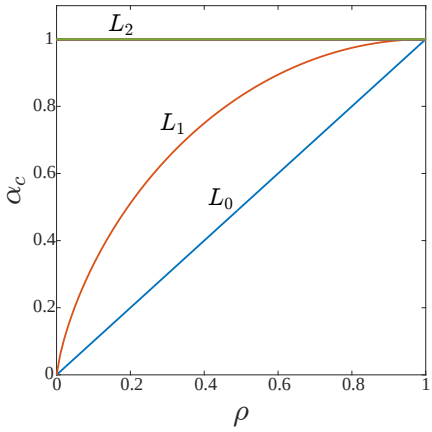


Figure 2.2.1: Phase diagram of the space of solutions of the CS problem.

is the most interesting limit. The case of the L_2 regularization is instead trivial as we need $\alpha \geq 1$ and so $M \geq N$ to reach perfect reconstruction of the signal; in this limit it suffices to trivially solve (if $\mathbf{F}$ is a full rank matrix) $\boldsymbol{x} = \mathbf{F}^{-1}\boldsymbol{y}$ which is not a compressed sensing problem.

It has been proved [26, 9] that the limit of perfect reconstruction of the Belief Propagation algorithm (see section 3.4.1) is delimited by a curve that lies in a region in between the L_1 and L_0 curves. Results employing the Belief Propagation algorithm and shown in chapter 6 are consistent with this limit.

2.2.2 The space of solution of constrained metabolic fluxes

Living organisms perform a never-ending stream of chemical reactions to maintain an environmental condition favorable to their self-sustenance [31, 32]. The elementary units responsible for this dynamic equilibrium are the cells. Within a cell, molecules like aminoacids, sugars, nucleotides, and lipids are processed by a huge set of enzymes to supply energy, to build proteins and other macro-molecules necessary to the cell life. All these complicated mechanisms take the name of *metabolism*.

The entire sets of possible reactions and all the molecules (often called metabolites) entering in these processes are well known from chemistry. Often reactions occur one after the other in the sense that the products of a reaction are the starting materials for other reactions and so on. The "velocity" of reaction, i.e. the rate at which each reaction takes place, is not known *a priori*. We define this rate as *metabolic flux*.

More formally, we can associate with each reaction a linear equation that describes the number of metabolites consumed or produced; the proportionality factors are called

stoichiometric coefficients. If we consider all possible reactions, we can build a matrix of these coefficients: the stoichiometric matrix. Let us consider a metabolic network of M metabolites and N reactions characterized by a $M \times N$ stoichiometric matrix $\mathbf{S}$, where the entry S_{ij} quantifies how metabolite i is produced (degraded) in reaction j; notice that typically $N > M$. Let us study how the concentrations $\boldsymbol{c}$ of metabolites change in time. Assuming that the concentration of metabolites are conserved, we can write the following continuity (or conservation) equation

$$\frac{\partial \boldsymbol{c}}{\partial t} = \mathbf{S}\boldsymbol{\nu} - \boldsymbol{o} + \boldsymbol{i} \tag{2.2.10}$$

where $\boldsymbol{\nu} = \{\nu_i\}_{i=1,\dots,N}$ are the metabolic fluxes, $\boldsymbol{i}$ and $\boldsymbol{o}$ are respectively the input and output metabolic flows. Now if we define $\boldsymbol{i} - \boldsymbol{o} \equiv \boldsymbol{b} = \{b_i\}_{i=1,\dots,M}$ as the net metabolic uptake and we investigate the system at steady state condition, we obtain the linear system of equations

$$\mathbf{S}\boldsymbol{\nu} = \boldsymbol{b} \tag{2.2.11}$$

where the fluxes $\boldsymbol{\nu} \in \mathbb{R}^N$ have to be determined. Notice that the relation in (2.2.11) cannot be inverted being the system of equations ill-posed; in principle there exists infinite configurations of fluxes satisfying (2.2.11). Fortunately, fluxes cannot take any possible value, are instead bounded, that is

$$\boldsymbol{\nu}^{min} \leq \boldsymbol{\nu} \leq \boldsymbol{\nu}^{max} \tag{2.2.12}$$

where $\boldsymbol{\nu}^{min}$ and $\boldsymbol{\nu}^{max}$ are the minimum and maximum allowed values of the fluxes. Bounds enforce some physiological constraints: on irreversible reactions we must impose $\nu \geq 0$, or, for those fluxes that are fixed to a certain value for maintenance reasons, we need to impose small interval of variation. In all other cases bounds are computed through the so-called Michaelis-Menten equation or set to very large values if no information is available. The solution space is thus bounded and it can now be described by a convex polytope in a $N - M$ dimensional space; any point within the polytope is an allowed configuration of fluxes. We will show several approaches to investigate the space of solutions of this problem. The first one is Flux Balance Analysis (see section 3.6.1.3) that consists in finding the configurations of fluxes (a point within the polytope) such that the growth rate of the cell is maximized; the second one is a sampling technique called Hit-and-Run Monte Carlo (see section 3.6.2.1) able to estimate numerically the marginal probability of each flux. In chapter 7 we present an analytic approach able to accurately

approximate the marginal probability density of metabolic fluxes. The approximation relying on the Expectation Propagation algorithm is presented in section §3.5.

Apart from the stoichiometric constraints affecting metabolic fluxes, we mention that the general process is simultaneously driven by regulatory constraints on enzymes participating to the metabolism, their corresponding genes and genes transcription factors [33]. However, integrating information on regulatory processes dramatically affects the complexity of the model and so far, theoretical research basically focuses its attention on (simple) stoichiometric models which, nonetheless, provides a significant explanation of cell functioning.

2.2.2.1 Fixing experimental profiles of fluxes

Let us define a slightly modified problem involving metabolic fluxes. Consider an organism which is well modeled by constraints (2.2.11) and (2.2.12) and let us suppose of having an experimental evidence of the *distribution* of a certain flux ν_i. How the metabolism of this organism has changed to accommodate this flux profile? We show in section §7.3 how to cope to this specific problem using the Expectation Propagation algorithm.

2.2.3 An inverse problem in electro-encephalography

The electrical activity of the brain is detected through electro-encephalograph (EEG) recordings which experimentally provide measures of voltage on the scalp. These voltages are the direct consequence of the presence of current densities within the brain. From a physical viewpoint, we can relate the measurements at time t to the current density through

$$v\left(\boldsymbol{r}_s, t\right) = \int d\boldsymbol{r}_g K\left(\boldsymbol{r}_s, \boldsymbol{r}_g\right) j\left(\boldsymbol{r}_g, t\right) \tag{2.2.13}$$

where $\boldsymbol{r}_s$ refers to a point on the scalp and $\boldsymbol{r}_g$ span the space covered by g^{th} source; $j\left(\boldsymbol{r}_g, t\right)$ is the current at point $\boldsymbol{r}_g$ at time t while $K\left(\boldsymbol{r}_s, \boldsymbol{r}_g\right)$ is the kernel matrix (or electric lead field) that summarizes the electric and geometrical properties of the media (brain, skull and scalp).

The inverse problem of locally identifying the sources of the currents, given a set of measurements of voltages, corresponds to the solution of a Fredholm integral equation of the first kind, or, using an inference jargon, the inverse problem of the EEG [34]. Even at steady state condition, solving this problem is very hard due to the ill-posed nature of the problem; in fact, many distributions for the currents satisfy (2.2.13). To deal with it, we will introduce a more simplified model for the static case. We underline that solving this inverse problem is of huge impact in medical diagnosis as it provides an accurate

knowledge of the current distributions within the brain through non-invasive medical devices.

Let us model the brain as a 3D "sphere" in a discretized space where voltages are now the solutions of a set of linear equations involving the (discrete) electric lead field and current densities. Formally, we group the voltage records in a vector $\boldsymbol{v}$ of N_e components, where N_e is the number of the electrodes (typically $N_e \in [32, 64]$). In our discrete model of the brain activity, the vector $\boldsymbol{v}$ is given by:

$$\boldsymbol{v} = \mathbf{K}\boldsymbol{j} + \boldsymbol{\varepsilon} \tag{2.2.14}$$

where $\mathbf{K}$ is the kernel matrix of dimension $N_e \times N_g$, N_g is the number grid volumes (called voxels) in the discretized brain, $\boldsymbol{j}$ is a vector of dimension N_g containing the current densities in each point of the space and $\boldsymbol{\varepsilon}$ models the additive noise affecting the measures. The problem of inferring the current densities from given voltage measures and a given electric lead field $\mathbf{K}$ is as hard as in the continuous case being the system of equations in (2.2.14) under-determined. Notice in fact that $N_g \gg N_e$ (typically $N_g \sim 10^3$) and thus the solution is not unique. Moreover, addressing the pseudo-inverse problem, that is finding $\boldsymbol{j} = \left(\mathbf{K}^T\mathbf{K}\right)\mathbf{K}^T \cdot (\boldsymbol{v} + \boldsymbol{\delta})$ where $\boldsymbol{\delta}$ is a small perturbation of the voltages, is not sufficient as solutions $\boldsymbol{j}$ strongly depend on the choice of $\boldsymbol{\delta}$.

One way of reducing the space of solutions is to impose certain additional constraints on $\boldsymbol{j}$. A convenient choice could be to assume sparsity on the components of $\boldsymbol{j}$ and to map this problem into a compressed sensing problem as described in section §2.2.1. However, these sparse solutions would correspond to point-like activation regions which is not what biologically evidences predict. What we seek are "sparse" but also "smooth" solutions, in the sense that current densities must spread in few volumes of the brain and than smoothly decay to zero in neighbor non-involved areas. The active regions correspond to well known areas of the brain those subdivision provides a map of the cerebral cortex according to neural organization. Thus, the constraints on the unknown variables must be carefully designed to impose sparseness, smoothness and correlate the components of $\boldsymbol{j}$. We propose in chapter 8 an algorithm based on Expectation Propagation approximation able to spatially localize the sources of synthetic currents within brain.

2.2.4 Inference in tomographic images

Tomography is an imaging technique of utmost importance in diagnostic medicine [35, 36]. It consists in illuminating an object (in medicine, an internal organ) from different directions and detecting transmitted or reflected rays. From the set of measurements

(often called projections), it reconstructs *a posteriori* the 2D cross-section of the object. The first medical applications used X-rays but, nowadays, radioisotopes and ultrasounds are successfully employed. Physically, the projections are the result of the energy transmitted (or reflected) by the object when illuminated.

If we performed a large set of measurements, we would be able to perfect reconstruct the object but, due to some practical issues, like the (usual) large size of the objects or the limited scanning directions of the devices, the rays (projections) are emitted (collected) in a limited set of angles. This affects the goodness of the tomographic images and efficient reconstruction algorithms are needed. In addition, it is preferable to use the least possible number of rays not to expose the patient (in diagnostic medicine) to excessive radiation that, even in small quantity, can damage the tissues.

The first attempt to reconstruct a tomographic image comes from the use of the Radon anti-transform that we will briefly outline here. Mathematically speaking, a projection at a given angle is the integral of the image in the direction of the ray. Let us define $f(x, y)$ the "intensity" of the image at position (x, y), and $t = x\cos\theta + y\sin\theta$ the parametrization of the line covered by rays emitted at angle θ . The device will detect

$$y_{\theta,t} = \int\int_{-\infty}^{+\infty} dxdy f(x, y)\,\delta(x\cos\theta + y\sin\theta - t) \tag{2.2.15}$$

where the function $y_{\theta,t}$ is known as the Radon transform of the function $f(x, y)$ [35]. Notice that performing its anti-transform will solve the inverse problem.

What we will present in this work is typically applied to X-rays computed tomography (CT) where the intensity of the images is determined by the photon absorption of the medium. Let us partition the image in $L \times L$ grid of $N = L^2$ pixels, where each pixel x_i for $i \in \{1, \ldots, N\}$ is treated as a continuous variable and let us call $\boldsymbol{y}$ the set of M measurements. The Beer-Lambert law of photons absorption connects the measurements with the intensities $\boldsymbol{x}$ as

$$\boldsymbol{y} = \mathbf{F}\boldsymbol{x} + \boldsymbol{w} \tag{2.2.16}$$

where $\mathbf{F}$ is a $M \times N$ tomographic projection matrix whose entries F_{ij} take value 1 if pixel j has been illuminated by beam i and 0 otherwise. Another possible definition of $\mathbf{F}$ requires that each element F_{ij} is the length of the portion of ray i passing through pixel j. The vector $\boldsymbol{w}$ contains the additive noise of the measurements. Typically the number of measurements is equal to $M = L \times n_\theta$ where n_θ is the number of directions of the emitted beams. Notice that the linear system of equations in (2.2.16) is ill-posed. Clearly, the $\boldsymbol{x}$ variables must satisfy additional constraints. If we are dealing with true detectors we know that under (or over) a certain amount of intensity it is not possible

to detect the signal (or intensity saturates). So, the value of each component x_i should be bounded in an interval $[x_i^{min}, x_i^{max}]$ where, for instance, $x_i^{min} = 0$ (or black, in a gray scale) and $x_i^{max} = 1$ (or white). Most importantly, as images represent "something", in the sense that the values of the intensities are not random, neighbors pixels should be correlated. We show in chapter 9 how to implement an Expectation Propagation algorithm to face this problem.

Chapter 3

Methods

This chapter provides a detailed description of the methodologies used to face the problems in chapter 2; these approaches fall under the class of variational methods. To be precise we will present the Belief Propagation algorithm for discrete and continuous variables, the Max Sum algorithm (the zero-temperature limit of the Belief Propagation equations for discrete variables) and the Expectation Propagation algorithm. The purpose of all the approaches that we will outline here is to approximate intractable probability distributions that describe the variables of the problems we are dealing with. We underline that, using the mapping introduced in chapter 1, they will have the functional form of Boltzmann-Gibbs distributions. The methods for treating discrete variables will be used in Part II to solve the variants of the Steiner tree problem while, for all the inference problems in chapter 2, we will apply continuous variables based tools. In particular, we will exploit here the compressed sensing problem as a guideline to the introduction of the Belief Propagation equations for continuous variables and of the Expectation Propagation equations.

At the end of the chapter we mention some other techniques used to solve the problems in chapter 2 whose results, in most of the cases, will be compared in Part II to our results.

3.1 Bayesian inference methods

We have introduced in section §1.3 Bayes' theorem from which we can design *a posteriori* the probability distribution of the variables we want to infer. Once this distribution is defined, there exist several approaches to estimate the values of the unknowns from their joint distribution. We show here three of them, the maximum a posteriori, the maximum likelihood and the minimum mean square error estimator. The last one will be used in almost all the inference problems we will face in this thesis.

- **Maximum a posteriori** (MAP). In this approach the predicted value of $\boldsymbol{x}$ is the most probable one according to the posterior probability, that is

$$\boldsymbol{x}_{MAP} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{y}) \tag{3.1.1}$$

- **Maximum likelihood estimation** (MLE). When we do not have any prior knowledge on the $\boldsymbol{x}$, the prior is set to an uniform distribution over the space of configurations of $\boldsymbol{x}$; in this case the posterior is not well defined. However, we can still maximize the likelihood as

$$\boldsymbol{x}_{MLE} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{y}|\boldsymbol{x}) \tag{3.1.2}$$

- **Minimum mean square error** (MMSE). Differently to the first two approaches that address the problem of finding the most probable value for $\boldsymbol{x}$, one may ask what are the values of the x_i such that the mean square error between each unknown and the corresponding true variable is minimized. One can prove that this is equivalent to compute

$$x_i^{MMSE} = \int dx_i x_i P(x_i|\boldsymbol{y}) \tag{3.1.3}$$

  where

$$P(x_i|\boldsymbol{y}) = \int \prod_{j\neq i} dx_j P(\boldsymbol{x}|\boldsymbol{y}) \tag{3.1.4}$$

  is the marginal probability distribution. Thus our estimate is the first moment of the marginal posterior distribution.

Notice that the MAP estimator is precisely the ground state of a physical system characterized by the Hamiltonian $H(\boldsymbol{x}, \boldsymbol{y})$ defined in section 1.3.1 (it is a global minimum of the energy) while from a statistical physics viewpoint the mean defining MMSE estimator in (3.1.3) is, for the Ising model, the value of the magnetization, i.e. the average value of σ_i with respect to the Boltzmann distribution, for spin i. These estimators coincide in the $\beta \to +\infty$ limit of (1.3.2) but only when the optimum is unique.

However, the computation of the MMSE estimators would often require hard and impractical calculations. Let us consider the L_2 regularization presented in section 2.2.1.2: the posterior associated with this case is a multivariate Gaussian that can be efficiently marginalized and maximized. The MAP estimators coincide with the MMSE estimators and the predictions will be provided by the vector of the means of the posteriors. Differently, the marginalization of the posterior in (2.2.5) is unfeasible as it would require the hard computation of multidimensional integrals. As a consequence, one has to design

some approximation technique to estimate the exact posterior $P(\boldsymbol{x}|\boldsymbol{y})$. The rest of this section is devoted to introduce approximation techniques able to estimate intractable distribution of this kind. One of this, the Variational Bayes method, is outlined in example 5 while we discuss in detail how to use Belief Propagation in section 3.4.1 and Expectation Propagation algorithm in section §3.5 to the same purpose.

3.2 Variational methods

As already mentioned in section §1.1 in the canonical ensemble the free energy potential has a key role in the understanding of the behavior of a system at equilibrium. In fact, elementary units forming the system will configure in a way that the free energy is minimized. This important property can be equivalently stated as follows.

Let us consider a system defined in the gran-canonical ensemble distributed according to an unknown probability density. We define the Gibbs free energy, or variational free energy

$$\mathcal{F}[Q] = U[Q] - TS[Q] \tag{3.2.1}$$

for a certain distribution Q. We ask ourselves: "what is the distribution Q that describes the physical system at equilibrium?" From our physical intuition and if variational principle holds, the system will be described by the distribution Q^* that minimizes (3.2.1), that is

$$Q^* = \arg \min_{Q:\sum_{\boldsymbol{x}} Q(\boldsymbol{x})=1} \mathcal{F}[Q] \tag{3.2.2}$$

where the minimization is performed for all the distributions satisfying the normalization constraint. As standard, we can add a Lagrange multiplier to (3.2.1) enforcing the additional constraint as

$$\mathcal{F}[Q] = U[Q] - TS[Q] + \lambda \left[\sum_{\boldsymbol{x}} Q(\boldsymbol{x}) - 1 \right] \tag{3.2.3}$$

and minimize with respect to Q. It is easy to prove that the distribution that minimizes the variational free energy is the Boltzmann distribution and the Helmholtz free energy is just the Gibbs free energy computed at $P(\boldsymbol{x})$:

$$F = \mathcal{F}[P] \tag{3.2.4}$$

In fact, the variational free energy can be always written as

$$\mathcal{F}[Q] = F + D_{KL}[Q\|P] \tag{3.2.5}$$

where $D_{KL}[Q\|P]$ is the Kullback-Leibler divergence between the distribution Q and the Boltzmann distribution P. It is a measure of "distance" between two probability densities and its formal definition for discrete variables is

$$D_{KL}[Q\|P] \doteq \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \log \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x})} \tag{3.2.6}$$

This important observation is the starting point for a class of approximation techniques able to estimate the partition function (and thus the free energy) when its computation is impractical. This of course applies for both optimization and inference problems. The idea is to choose a trial set of suitable probability distributions $Q(\boldsymbol{x})$ (or, eventually, distributions $Q(\boldsymbol{x}|\boldsymbol{\theta})$ parametrized by $\boldsymbol{\theta}$) and associate with them a variational free energy as defined in (3.2.3). Hopefully, the Gibbs free energy computed at its stationary point is a good approximation of the intractable free energy. The most famous approximation methods used in statistical mechanics are the *mean-field* approximations that are closely related to the discussion of this section. More precisely, mean-field methods are characterized by the use of factorized trial probability distribution. We will show in example 4 the most simple mean-field approximation, the "naif" mean-field. Two more advanced mean-field techniques are presented in the rest of chapter 3: the Belief Propagation algorithm and the Expectation Propagation algorithm. Both approximations rely on a non-trivial factorization of the trial set of distributions.

Example 4. Naif mean-field

One of the simplest way of encompassing the ability of estimating the partition function of a statistical model, is to use the (naif) mean-field approximation. Here, mutual interactions among variables are approximated through the insertion of a fictitious external field applied to all of them. It results that variables are now independent but subjected to a sort of external interaction. Let us define the trial mean-field distribution Q_{MF} that, in this approximation, trivially factorizes as $Q_{MF}(\boldsymbol{x}) = \prod_i q_i(x_i)$. The unknown marginal distribution q_i are sought by minimization of the Gibbs free energy functional associated with Q_{MF}. The computation leads to a set of self-consistent equations involving the q_i to be solved iteratively.

This method is able to catch the main behavior of a system of spins (as the one in example 1) for instance it can predict phase transitions (for critical values of the

temperature or of the magnetic field) but it is not accurate on the estimate of correlations. To properly consider them one should deal with a more general set of trial probability distributions. For instance, the Belief Propagation algorithm (or Bethe-Peierls approximation technique) can be derived using a variational functional defined over more complicated trial probability densities as shown in section 3.3.3.1.

Example 5. Variational Bayes

When the approach proposed in section §3.2 aims at approximating a posterior distribution $P(\boldsymbol{x}|\boldsymbol{y})$, it takes the name of Variational Bayes (VB) method. Formally, (3.2.6) becomes

$$D_{KL}[Q\|P] = \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \log \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x}|\boldsymbol{y})} \tag{3.2.7}$$

$$= \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \log \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x},\boldsymbol{y})} - \log P(\boldsymbol{x}) \tag{3.2.8}$$

$$\propto \sum_{\boldsymbol{x}} [Q(\boldsymbol{x}) \log Q(\boldsymbol{x}) - Q(\boldsymbol{x}) \log P(\boldsymbol{x},\boldsymbol{y})] \tag{3.2.9}$$

3.3 Belief propagation

The Belief propagation (BP) method is an iterative algorithm developed independently in statistical physics as "Bethe-Peierls approximation" or "cavity methods", in computer science as "sum-product algorithm". It consists in a set of closed equations defined on a factor graph to be solved iteratively; at convergence it will provide an approximation of single-node and eventually of two-nodes marginals. Although this method is exact only on tree graph, it can be applied to general graphs giving a good estimate of marginals as long as the factor graph is locally tree-like [37]. In this case, long range correlations are negligible and BP is sufficient to catch the properties of the system.

Recently, BP algorithm (or variants of it) has been applied to a wide set of optimization and inference problems [1, 10, 38]. In the following we briefly outline the definition of a factor graph and then we present the message-passing algorithm following the approach in [1].

3.3.1 Preliminaries: factor graph representation

A *factor graph* is a graphical way of representing factorized probability distributions. Its purpose is essentially to represent interacting variables underlying their dependencies and the constraints they must satisfy. The main advantage is to allow for the application

of efficient algorithms, like the "sum-product" or "Belief Propagation" described in section §3.3.

A factor graph is a bipartite graph composed by two families of nodes:

- **Variable nodes**: They represent the variables under investigation. We know their support and their mutual dependencies;
- **Factor nodes**: They are the so-called compatibility functions involving, eventually, subsets of the variable nodes. They represent hard constraints to be satisfied by the variables.

Links of a factor graph either connect a factor node to their arguments, or connect variable nodes to the compatibility functions representing the constraints they must satisfy. For instance, let us consider the joint probability distribution of N variables $\boldsymbol{x} = \{x_1 \ldots, x_N\}$ as

$$P(\boldsymbol{x}) = \frac{1}{Z} \prod_{a=1}^{M} \psi_a(\boldsymbol{x}_{\partial a}) \tag{3.3.1}$$

where each of the M functions, $\{\psi_1, \ldots, \psi_M\}$, represents a constraint over the variables identified as $\boldsymbol{x}_{\partial a} = \{x_i | i \in \partial a\}$. In a factor graph the $\boldsymbol{x} = \{x_i\}_{i=1,\ldots,N}$ will be our variable nodes (often drawn as circles), instead the family of functions $\{\psi_a\}_{a=1,\ldots,M}$ will be the factor nodes (represented as square nodes).

Example 6. A first arborescence representation for the Steiner tree problem

Here we briefly explain how to build a factor graph representing the Boltzmann distribution of some auxiliary variables that allow us to map the Prize-collecting Steiner tree problem introduced in section 2.1.1 into a statistical mechanics problem. We present here the set of "pointer" variables of a factor graph, used in the works [20, 21], as an introduction to the the slightly different formalism based on "depth" variables presented in chapter 4

Consider a graph $\mathcal{G}(V, E)$ with weights $\boldsymbol{w}$ associated with edges, prizes $\boldsymbol{c}$ associated with profitable nodes. Let us associate with each node $i \in V$ of the original graph, a two components variable (p_i, d_i) where $p_i \in \{j : j \in \partial i\} \bigcup \{*\}$ and $d_i \in \{1, \ldots, D\}$. In a tree, p_i points to one of the neighbors of i (let us called it j), while d_i is the length of the unique path from node i to the root passing through p_i within the Steiner tree. The state $p_j = *$ conventionally means that i is not a member of the tree. To ensure that the configuration of the sets of variables $(\boldsymbol{p}, \boldsymbol{d})$ describe a tree, we must impose these two hard constraints: (i) if $p_i = j$ then $d_i = d_j + 1$, i.e. the distance decreases as we go closer to the root, (ii) if $p_i = j$ and j is not the root, then $p_j \neq *$.

Formally, we can define, for each couple of neighbors, a function

$$f_{ij} = 1 - \delta_{p_i,j}\left[1 - \delta_{d_i,d_j+1}\left(1 - \delta_{p_j,*}\right)\right] \tag{3.3.2}$$

that takes value 1 if both constraints are satisfied or 0 otherwise. Of course, the same holds in the opposite direction; thus, if $f_{ij} = 1$ ($f_{ij} = 0$) also $f_{ji} = 1$ ($f_{ji} = 0$). Let us build a factor graph having the same structure of $\mathcal{G}$ in which the original node are now variable nodes hosting the sets $(\boldsymbol{p}, \boldsymbol{d})$ and on each edge $(i,j) \in E$ we draw a factor node corresponding to the compatibility function $\psi_{(i,j)}(p_i, p_j, d_i, d_j) = f_{ij} f_{ji}$ equals to 1 if connectivity constraints are satisfied or 0 otherwise. Clearly, we can re-formulate the energy function in (2.1.1) as a function of our auxiliary variables. If we define an additional weight $w_{i*} = c_i$, an equivalent formulation of the energy reads

$$H(\boldsymbol{p}, \boldsymbol{d}) = \begin{cases} \sum_{i \in V} w_{ip_i} & \text{if } \prod_{(i,j)\in E} \psi_{ij}(p_i, p_j, d_i, d_j) = 1 \\ +\infty & \text{otherwise} \end{cases} \tag{3.3.3}$$

where we have set to infinity the energies of incompatible configurations. As explained in section 1.2.2, minimizing (3.3.3) is equivalent to determine the configuration of variables that maximizes

$$P_\beta(\boldsymbol{p}, \boldsymbol{d}) = \frac{1}{Z_\beta} e^{-\beta H(\boldsymbol{p}, \boldsymbol{d})} \tag{3.3.4}$$

in the $\beta \to +\infty$. It is possible to have an estimate of the marginals of this distribution applying Belief Propagation or Max Sum algorithm in the factor graph described here. We do not report here the details of the equations that are fully described in [20, 21].

3.3.2 Update equations

At each iteration t we associate with each edge (a, i) of the factor graph (where a is a factor node and i is a variable node) two "messages" $m^{(t)}_{i\to a}(x_i)$, from variable-to-factor, and $\hat{m}^{(t)}_{a\to i}(x_i)$, from factor-to-variable, flowing within the graph. They take values in the space of probability distributions over the single variable configuration space, that is, for the variable-to-factor message, $m^{(t)}_{i\to a} = \left\{m^{(t)}_{i\to a}(x_i) : x_i \in X\right\}$. It also must satisfy $m^{(t)}_{i\to a}(x_i) \geq 0$ and the normalization condition $\sum_{x_i} m^{(t)}_{i\to a}(x_i) = 1$. The same apply for $\hat{m}^{(t)}_{a\to i}$. Messages satisfied closed equations solved iteratively through a "local" update: "outgoing" messages (both from a variable and a factor node) will only depend on messages coming from its neighbors at the previews time-step. Formally, the update rules consist

in:

$$\begin{cases} m_{i\to a}^{(t+1)}(x_i) & = \frac{1}{Z_{i\to a}} \prod_{b\in\partial i\setminus a} \hat{m}_{b\to i}^{(t)}(x_i) \\ \hat{m}_{a\to i}^{(t+1)}(x_i) & = \frac{1}{\hat{Z}_{a\to i}} \sum_{\boldsymbol{x}_{\partial a\setminus i}} \psi_a(\boldsymbol{x}_{\partial a}) \prod_{j\in\partial a\setminus i} m_{j\to a}^{(t)}(x_j) \end{cases} \tag{3.3.5}$$

where $Z_{i\to a} = \sum_{x_i} m_{i\to a}^{(t+1)}(x_i)$ and $\hat{Z}_{a\to i} = \sum_{x_i} \hat{m}_{a\to i}^{(t+1)}(x_i)$. At $t \to +\infty$ messages converge to fixed point values $m_{i\to a}^{(\infty)}$ and $\hat{m}_{a\to i}^{(\infty)}$. From a probabilistic viewpoint, the former can be interpreted as the probability of variable i to assume value x_i in a modified graphical model where the factor a has been erased. Equivalently, the latter is in some sense the probability that node i assumes state x_i where all the neighboring factors except a have been deleted. Thus, we can approximate the marginal probability distribution for variable i considering all the incoming messages as

$$M_i(x_i) \propto \prod_{a\in\partial i} \hat{m}_{a\to i}^{(\infty)}(x_i) \tag{3.3.6}$$

3.3.3 Pairwise graphical model

If the system interacts only via two-bodies interactions, the factor graph reduces to a graph $\mathcal{G}(V, E)$ where each variable node is represented by a node $i \in V$ and factor nodes are just the links $(i, j) \in E$ connecting the nodes of the graph. The corresponding probability distribution reads

$$P(\boldsymbol{x}) = \frac{1}{Z} \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j) \tag{3.3.7}$$

In this scenario BP messages reduce to one family of messages, for instance the variable-to-factor set $m_{i\to(ij)}(x_i)$; for sake of simplicity we will simply denote it as $m_{i\to j}(x_i)$. The update rule becomes:

$$m_{i\to j}^{(t+1)}(x_i) \propto \sum_{\{x_k : k\in\partial i\setminus j\}} \prod_{k\in\partial i\setminus j} \psi_{ki}(x_i, x_k)\, m_{k\to i}^{(t)}(x_k) \tag{3.3.8}$$

In this framework it is easy to define the one-variable and two-variables marginals as

$$M_i(x_i) \propto \sum_{\{x_j : j\in\partial i\}} \prod_{j\in\partial i} \psi_{ij}(x_i, x_j)\, m_{j\to i}(x_j) \tag{3.3.9}$$

$$M_{ij}(x_i, x_j) \propto m_{i\to j}(x_i)\, m_{j\to i}(x_j) \tag{3.3.10}$$

3.3.3.1 Beyond naif mean-field

Belief propagation is equivalent to the Bethe-Peierls approximation of statistical physics [39]. The update equation in (3.3.8) can be formally derived from the minimization of a variational free energy, the Bethe free energy. Fixed points of BP algorithm therefore correspond to stationary points of the Bethe free energy.

The main feature of the Bethe-Peierls approximation is to assume that the joint probability in (3.3.7) can be approximated through a trial distribution $Q(\boldsymbol{x})$ expressed in terms of single-variable and two-variables marginals, or "beliefs", $b_i(x_i)$ and $b_{ij}(x_i, x_j)$ as

$$Q(\boldsymbol{x}) = \prod_{(i,j)\in E} \frac{b_{ij}(x_i, x_j)}{b_i(x_i)\, b_j(x_j)} \prod_i b_i(x_i) \tag{3.3.11}$$

The derivation of the message-passing equations is reported in appendix A.

3.3.4 Optimization and Max Sum algorithm

Very often the exact computation of marginals requires hard, and analytically unfeasible, computations; nevertheless we explained in section §3.3 how to properly approximate them using the "message-passing" algorithm on a factor graph. In the context of combinatorial optimization as well as in statistical inference we often face an assignment (or prediction) problem, that is we aim at determining what are the most probable states of the variables in interest. Given the approximate marginals in (3.3.6) our prediction, or our assignment, will be

$$x_i^* = \arg\max_{x_i} M_i(x_i) \tag{3.3.12}$$

Notice that in principle BP equations are defined at positive temperature, i.e. finite β. For optimization problems, we can simplify the BP equations computing explicitly the limit $\beta \to +\infty$: the algorithm that follows is known as Max-Sum (MS) algorithm. We underline that the limit is performed on the BP equations and therefore it is not guaranteed that the fixed point of MS algorithm corresponds to the ground state of the system, that is the zero-temperature limit of the associated distribution.

As a matter of example, let us consider the joint probability of a pairwise model in (3.3.7) and let us rewrite the compatibility functions in a Boltzmann factors fashion:

$$\psi_{ij}(x_i, x_j) = e^{-\beta H_{ij}(x_i, x_j)} \tag{3.3.13}$$

The messages update rule in (3.3.8) become

$$m_{i\to j}\left(x_i\right) \propto \sum_{\{x_k:k\in\partial i\setminus j\}} \prod_{k\in\partial i\setminus j} e^{-\beta H_{ik}(x_i,x_k)} m_{k\to i}\left(x_k\right) \tag{3.3.14}$$

Let us define the set of messages in the $\beta\to+\infty$ as $h_{i\to j}\left(x_i\right) = \lim_{\beta\to+\infty}\frac{1}{\beta}\ln m_{i\to j}\left(x_i\right)$. Changing the variables in (3.3.14), we get

$$\begin{aligned}\frac{1}{\beta}\ln m_{i\to j}\left(x_i\right) &\propto \frac{1}{\beta}\ln \prod_{k\in\partial i\setminus j}\sum_{\{x_k:k\in\partial i\setminus j\}} e^{-\beta H_{ik}(x_i,x_k)} m_{k\to i}\left(x_k\right) \\ &\propto \frac{1}{\beta}\sum_{k\in\partial i\setminus j}\ln \sum_{\{x_k:k\in\partial i\setminus j\}} e^{-\beta H_{ik}(x_i,x_k)+\ln m_{k\to i}(x_k)}\end{aligned} \tag{3.3.15}$$

If now we perform the $\beta\to+\infty$, the sum in (3.3.15) reduces to the dominant term

$$e^{\max_{\{x_k:k\in\partial i\setminus j\}}[-\beta H_{ik}(x_i,x_k)+\ln m_{k\to i}(x_k)]}$$

and therefore

$$h_{i\to j}\left(x_i\right) = \max_{\{x_k:k\in\partial i\setminus j\}} \sum_{k\in\partial i\setminus j}\left[-H_{ik}\left(x_i,x_k\right)+h_{k\to i}\left(x_k\right)\right] - C \tag{3.3.16}$$

where C is an additive constant ensuring that the normalization condition, that is $\max_{x_i} h_{i\to j}\left(x_i\right) = 0$, is satisfied. The expression in (3.3.16) justifies the name "Max-Sum"; in fact, one can easily derive this algorithm just replacing a "sum" in BP equations with a "max" and a "product" with a "sum". These messages take value in the interval $(-\infty, 0]$.

We can also define the $\beta\to+\infty$ equivalent of the marginal in (3.3.9) and (3.3.10) as

$$\begin{aligned}M_i^{\beta\to+\infty}\left(x_i\right) &\propto \sum_{k\in\partial i}\max_{x_k}\left[-H_{ik}\left(x_i,x_k\right)+h_{k\to i}\left(x_k\right)\right] \\ M_{ij}^{\beta\to+\infty}\left(x_i,x_j\right) &\propto h_{i\to j}\left(x_i\right)+h_{j\to i}\left(x_j\right)+H_{ij}\left(x_i,x_j\right)\end{aligned} \tag{3.3.17}$$

3.4 Belief propagation for continuous variables

The Belief Propagation algorithm described in section §3.3 deals with the approximation of marginal probability distributions defined over discrete variables. A generalization to continuous variables is possible; the algorithm that follows is known as relaxed Belief Propagation (r-BP) [40, 26]. Briefly, BP messages are now difficult to compute but, under the decorrelation hypothesis (or eventually weak correlation hypothesis) behind

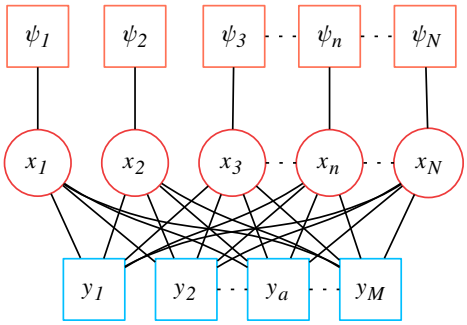


Figure 3.4.1: Factor graph associated with the distribution in (3.4.1)

the approximation, they can be approximated as Gaussian probability densities. In the following we will present this algorithm applied to the compressed sensing problem described in section 2.2.1.1.

3.4.1 BP equations for the CS problem

For sake of simplicity we rewrite (2.2.5) as

$$P(\boldsymbol{x}|\boldsymbol{y}) \propto \prod_{a=1}^{M} P_a(y_a|\boldsymbol{x}) \prod_{n=1}^{N} \psi_n(x_n) \tag{3.4.1}$$

where P_a is the compatibility function relative to each a-factor of the likelihood, that is $P_a(y_a|\boldsymbol{x}) = e^{-\frac{1}{2\Delta}\left(y_a - \sum_n F_{an}x_n\right)^2}$; notice that it is function of all the $\boldsymbol{x}$. The constraint ψ_n enforces the sparsity prior according to the L_0 regularization, $\psi_n(x_n) = (1-\rho)\,\delta(x_n) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{x_n^2}{2\lambda}}$, applied to each single component.

Let us associate a factor graph with the posterior probability defined in (3.4.1). Each of the factor nodes $\{y_a\}_{a=1,\dots,M}$ is connected to all variables nodes $\{x_i\}_{i=1,\dots,N}$ in a fully connected graph whose edges are weighted according to P_a. Instead, factors associated with each priors ψ_n are in one-to-one correspondence with variable nodes $\{x_i\}_{i=1,\dots,N}$. The sketch is reported in figure 3.4.1.

Let us define two families of messages: a factor-to-variable $m_{a\to n}(x_n)$ and a variable-to-factor $m_{n\to a}(x_n)$ passing through the fully connected graph in figure 3.4.1. Formally, they satisfy

$$\begin{cases} m_{a\to n}(x_n) = \frac{1}{Z_{a\to n}} & \int \prod_{m\neq n} dx_m m_{m\to a}(x_m)\, P_a(y_a|\boldsymbol{x}) \\ m_{n\to a}(x_n) = \frac{1}{Z_{n\to a}} & \int dx_n \psi_n(x_n) \prod_{b\neq a} m_{b\to n}(x_n) \end{cases} \tag{3.4.2}$$

where $Z_{a\to n} = \int dx_n m_{a\to n}(x_n)$ and $Z_{n\to a} = \int dx_n m_{n\to a}(x_n)$. At difference to the discrete case, the direct computation of (3.4.2) is intractable since for each variable node we need to perform an $N-1$ dimensional integral; nonetheless some approximations can be performed.

Let us compute, for instance, the first term in (3.4.2)

$$m_{a\to n}(x_n) \propto \int \prod_{m\neq n} dx_m m_{m\to a}(x_m)\, e^{-\frac{1}{2\Delta}\left(\sum_{m\neq n} F_{am}x_m + F_{an}x_n - y_a\right)^2} \tag{3.4.3}$$

$$= \int dS e^{-\frac{(S+F_{an}x_n-y_a)^2}{2\Delta}} \int \prod_{m\neq n} dx_m \delta\left(S - \sum_{m\neq n} F_{am}x_m\right) m_{m\to a}(x_m) \tag{3.4.4}$$

The integral in (3.4.4) is the convolution of the messages that can be interpreted as the probability of observing the weighted sum $\sum_{m\neq n} F_{am}x_m$ when the marginal probability of each variable x_m for $m \neq n$ is given by $m_{m\to a}(x_m)$. If the terms of this sum are statistically uncorrelated, as the BP assumption states, we can use the Central Limit Theorem (CLT) and approximate this probability with a Gaussian probability density of mean $S^{\mu}_{a\to n} = \sum_{m\neq n} F_{am}\mu_{m\to a}$ for $\mu_{m\to a} = \int dx_m x_m m_{m\to a}(x_m)$ and variance $S^{\sigma}_{a\to n} = \sum_{m\neq n} F^2_{am}\sigma^2_{m\to a}$ for $\sigma^2_{m\to a} = \int dx_m (x_m - \mu_{m\to a})^2 m_{m\to a}(x_m)$. Thus

$$m_{a\to n}(x_n) \propto \int dS e^{-\frac{1}{2\Delta}(S+F_{an}x_n-y_a)^2} e^{-\frac{(S-S^{\mu}_{a\to n})^2}{2S^{\sigma}_{a\to n}}} \tag{3.4.5}$$

Integrating the Gaussian distribution in (3.4.5), we obtain

$$m_{a\to n}(x_n) = \sqrt{\frac{A_{a\to n}}{2\pi}} e^{-\frac{B^2_{a\to n}}{2A_{a\to n}}} e^{-\frac{A_{a\to n}}{2}x_n^2 + B_{a\to n}x_n} \tag{3.4.6}$$

where

$$A_{a\to n} = \frac{F^2_{an}}{(\Delta + S^{\sigma}_{a\to n})} \qquad B_{a\to n} = \frac{F_{an}(y_a - S^{\mu}_{a\to n})}{(\Delta + S^{\sigma}_{a\to n})} \tag{3.4.7}$$

To close the equations we must specify the values for the set of the messages $m_{n\to a}(x_n)$, the mean $\mu_{n\to a}$ and variance $\sigma^2_{n\to a}$ of the Gaussian message in (3.4.1); the derivation and the explicit expressions are reported in appendix B.

This algorithm relies on the decorrelation assumption among variables that is even underlined by the use of CLT in the convolution in (3.4.4). This is satisfied when the rows of the matrix $\mathbf{F}$ are uncorrelated, for instance when $\mathbf{F}$ is a random matrix. Although this is not the case in general setting, we can still use this algorithm keeping in mind that the performances strongly depend on how much this hypothesis holds. For

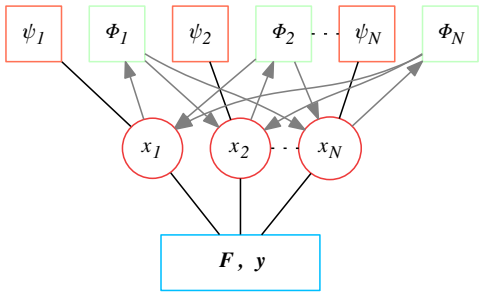


Figure 3.5.1: Picture of the EP marginalization for CS problem

low-correlated matrices, it may be applied achieving good performances but if entries are strong correlated, for instance in structured matrices, BP often does not converge. We will show some examples in section §6.3.

3.5 Expectation propagation

Expectation Propagation (EP) is an efficient algorithm introduced by Minka [41] in the framework of Bayesian inference and by Opper et al. in statistical physics with the name of Expectation Consistent [42–44]. EP is an iterative algorithm able to approximate intractable and factorized probability distributions.

Here we present the Expectation Propagation algorithm from a new point of view. Following the statistical inference approach for the compressed sensing problem, we present EP as a way of overcoming the limitation of BP algorithm described in section 3.4.1. A standard derivation of the EP algorithm is presented in appendix C.

3.5.1 EP algorithm for linear estimation problems

Consider the general setting of the CS problem described in section 2.2.1.1 and the factor graph in figure 3.4.1. BP algorithm relies on a Gaussian approximation of the messages flowing on a fully connected graph. This choice is certainly advantageous for the computation of marginals, but it can be applied to a restrict set of instances.

Our idea is to (i) relax the hypothesis of weak correlation among $\boldsymbol{x}$ and thus among the entries of $\mathbf{F}$ and (ii) keeping a Gaussian approximation to preserve the feasibility in computing the marginals. One way of treating this problem is to use a Gaussian approximation directly applied to prior factors in (3.4.1).

Consider the picture in figure 3.5.1 where we have added N "factor nodes" [1] of the type

$$\phi_n(x_n) \equiv \frac{1}{\sqrt{2\pi b_n}} e^{-\frac{(x_n-a_n)^2}{2b_n}} \qquad n = 1, \ldots, N \tag{3.5.1}$$

parametrized by the mean a_n and variance b_n. Each new function ϕ_n represents an approximation of the exact prior $\psi_n(x_n)$ that is connected by an incoming arrow to its argument x_n and via outgoing arrows to all the other variables $\{x_m\}_{m\neq n}$. Let us define our estimate of the marginal probability distribution of variable x_n as $Q^{(n)}(x_n)$. Consistently to the "message-passing" viewpoint and the graph in figure 3.5.1, it will read

$$Q^{(n)}(x_n|\boldsymbol{y}) \propto \int \prod_{m\neq n} dx_m P(\boldsymbol{y}|\boldsymbol{x})\, \psi_n(x_n) \prod_{m\neq n} \phi_m(x_m) \tag{3.5.2}$$

where we have not specified the normalization constant. The argument of the integral in usually called *tilted* or *leave-one-out* distribution of the n^{th} variable as to underline its dependency on the entire set of Gaussian distributions except for the n^{th} one that it is replaced with the associated exact prior $\psi_n(x_n)$. Formally, it is defined as

$$Q^{(n)}(\boldsymbol{x}|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{x})\, \psi_n(x_n) \prod_{m\neq n} \phi_m(x_m) \tag{3.5.3}$$

The goodness of this approximation strongly depend on how we determine the set of parameters characterizing the Gaussian terms, that are the means $\boldsymbol{a}$ and the variances $\boldsymbol{b}$. In the following we describe how EP algorithm provide these parameters. Let us define the *full* approximation of the posterior as the one containing all the Gaussian priors

$$Q(\boldsymbol{x}|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{x})\, \phi_n(x_n) \prod_{m\neq n} \phi_m(x_m) \tag{3.5.4}$$

Of course we can compute a marginal probability for the variable x_n as

$$Q(x_n|\boldsymbol{y}) \propto \int \prod_{m\neq n} dx_m P(\boldsymbol{y}|\boldsymbol{x})\, \phi_n(x_n) \prod_{m\neq n} \phi_m(x_m) \tag{3.5.5}$$

We expect that the estimate of the marginal in (3.5.3) will be more accurate than the one in (3.5.5) since in the tilted distribution the prior associated with the variable in interest is treated exactly. If this observation is true, and we assume it is, we can use it as a criterion to determine the parameters of approximate prior $\phi_n(x_n)$ associated with the

[1] Notice that the graph in figure 3.5.1 has not to be interpreted as a factor graph due to the presence of directed edges. It has the purpose of visualizing in a graph the marginalization of each variable according to EP algorithm, as in (3.5.2).

variable x_n. For each n^{th} variable, we can ask ourselves: "how can we assign the values of a_n and b_n such that the marginal in (3.5.5) is as similar as possible to the marginal in (3.5.3)?" To answer this question, one can compute a measure of the "distance" between the two distributions and minimize it with respect to the two parameters. In the framework of probability theory and information theory, such measure is given by the Kullback-Leibler divergence for continuous variables that, according to the definition in (3.2.6) , is given by

$$D_{KL}\left[Q^{(n)} \| Q\right] = \int d\boldsymbol{x} Q^{(n)}\left(\boldsymbol{x}|\boldsymbol{y}\right) \log \frac{Q^{(n)}\left(\boldsymbol{x}|\boldsymbol{y}\right)}{Q\left(\boldsymbol{x}|\boldsymbol{y}\right)} \tag{3.5.6}$$

Thus we seek at computing

$$(a_n^*, b_n^*) = \arg \min_{(a_n, b_n)} D_{KL}\left[Q^{(n)} \| Q\right] \tag{3.5.7}$$

As proven in appendix D, it is not surprising that the minimization in (3.5.7) is equivalent to match the first and second moments of $Q^{(n)}\left(x_n|\boldsymbol{y}\right)$ and $Q\left(x_n|\boldsymbol{y}\right)$. Formally:

$$\begin{cases} \langle x_n \rangle_{Q^{(n)}} & = \langle x_n \rangle_Q \\ \langle x_n^2 \rangle_{Q^{(n)}} & = \langle x_n^2 \rangle_Q \end{cases} \tag{3.5.8}$$

Notice that both the distributions in (3.5.3) and (3.5.4) have a common factor: the likelihood times $N-1$ Gaussian priors. Since for the CS problem as well as for all the models described in Part II the likelihood function will take the form of a Gaussian distribution, the common factor can be written as a multivariate Gaussian. Let us group all the $N-1$ approximate priors in (3.5.3) and (3.5.4) into a multivariate Gaussian distribution as

$$\prod_{m \neq n} e^{-\frac{(x_m - a_m)^2}{2b_m}} = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{a})^T \mathbf{D}_{(n)} (\boldsymbol{x}-\boldsymbol{a})} \tag{3.5.9}$$

where $\mathbf{D}_{(n)}$ is a diagonal matrix whose elements are $D_{mm} = \frac{1}{b_m}$ for $m \neq n$ and $D_{nn} = 0$ (of course out of diagonal elements are zeros). Grouping together the $\boldsymbol{x}$-dependent terms of the argument of the exponential in (3.5.3) and (3.5.4) we obtain

$$P\left(\boldsymbol{y}|\boldsymbol{x}\right) \prod_{m \neq n} \phi_m\left(x_m\right) \propto e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)^T \boldsymbol{\Sigma}_{(n)}^{-1} \left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)} \tag{3.5.10}$$

where $\boldsymbol{\Sigma}_{(n)}^{-1}$ is an inverse covariance matrix and $\boldsymbol{\mu}^{(n)}$ is a vector of averages that depend on the n^{th} variable we are updating. Thus the first and second moments in (3.5.8) will

always take the form of

$$\langle x_n^\alpha \rangle_{Q^{(n)}} \propto \int dx_n x_n^\alpha e^{-\frac{(x_n-\mu_n)^2}{2\Sigma_{nn}}} \psi_n(x_n) \tag{3.5.11}$$

$$\langle x_n^\alpha \rangle_{Q} \propto \int dx_n x_n^\alpha e^{-\frac{(x_n-\mu_n)^2}{2\Sigma_{nn}}} \phi_n(x_n) \tag{3.5.12}$$

where $\alpha = \{1, 2\}$ and μ_n is the n^{th} element of $\boldsymbol{\mu}^{(n)}$ and $\Sigma_{nn} = \left(\boldsymbol{\Sigma}_{(n)}\right)_{nn}$. Contrary to what happens to the moments of the tilted distribution in (3.5.11) that strongly depend on the form of the exact prior, the mean and the variance of the full Gaussian approximation in (3.5.12) always give

$$\begin{cases} \langle x_n \rangle_Q & = \left(\frac{1}{b_n} + \frac{1}{\Sigma_{nn}}\right)^{-1} \left(\frac{a_n}{b_n} + \frac{\mu_n}{\Sigma_{nn}}\right) \\ \langle x_n^2 \rangle_Q - \langle x_n \rangle_Q^2 & = \left(\frac{1}{b_n} + \frac{1}{\Sigma_{nn}}\right)^{-1} \end{cases} \tag{3.5.13}$$

If we insert the expressions in (3.5.13) inside (3.5.8) one finds the EP update equations in the most general framework:

$$\begin{cases} b_n & = \left(\frac{1}{\langle x_n^2 \rangle_{Q^{(n)}} - \langle x_n \rangle_{Q^{(n)}}^2} - \frac{1}{\Sigma_{nn}}\right)^{-1} \\ a_n & = b_n \left[\langle x_n \rangle_{Q^{(n)}} \left(\frac{1}{b_n} + \frac{1}{\Sigma_{nn}}\right) - \frac{\mu_n}{\Sigma_{nn}}\right] \end{cases} \tag{3.5.14}$$

The parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ are iteratively updated until we reach a fixed point, or, in other words, EP numerically converges. At each iteration t we compute an error ε which measures how the approximate marginal distributions change in two consecutive iterations. Formally, we define the error as the maximum value of the sum of the differences (in absolute values) of the mean and second moment of the marginal distribution, that is

$$\varepsilon^t = \max_n \left| \langle x_n \rangle_{Q^{(n)}}^{t+1} - \langle x_n \rangle_{Q^{(n)}}^{t} \right| + \left| \left\langle x_n^2 \right\rangle_{Q^{(n)}}^{t+1} - \left\langle x_n^2 \right\rangle_{Q^{(n)}}^{t} \right| \tag{3.5.15}$$

If ε^t is smaller than a predetermined precision (for instance 10^{-5}), the iteration stops.

3.5.2 Divergence measures

Expectation Propagation algorithm is more than a heuristic method to estimate intractable distributions; we explain here its link to variational techniques as explained in [45].

As we have outlined in section 1.3.1 VB techniques aim at analytically seeking a tractable approximation $Q(\boldsymbol{\nu}|\boldsymbol{b})$ to the exact posterior $P(\boldsymbol{\nu}|\boldsymbol{b})$. This distribution is

determined requiring that the parameters characterizing Q are such that the Kullback-Leibler divergence $D_{KL}[Q \parallel P]$ is minimized. However, in many cases, due to the intractable form of $P(\boldsymbol{\nu}|\boldsymbol{b})$, this measure is not well defined. Differently, we could perform the minimization of a generalization of the Kullback-Leibler divergence, called α-divergence, that is defined, for continuous variables, as

$$D_\alpha(P\|Q) \equiv \frac{\int d^N\boldsymbol{x}\alpha P(\boldsymbol{x}) + (1-\alpha)Q(\boldsymbol{x}) - P(\boldsymbol{x})^\alpha Q(\boldsymbol{x})^{1-\alpha}}{\alpha(1-\alpha)} \tag{3.5.16}$$

where for $\alpha \to 0$ we recover $D_{KL}[Q \parallel P]$ and for $\alpha \to 1$ we obtain $D_{KL}[P \parallel Q]$. Also minimizing $D_{KL}[P \parallel Q]$ is impractical as we should compute averages with respect to $P(\boldsymbol{\nu}|\boldsymbol{b})$. In this perspective EP algorithm tries to perform at each iteration, a "local" minimization of the α-divergence (for $\alpha \to 1$) in the sense that the "exact" considered distribution is not $P(\boldsymbol{\nu}|\boldsymbol{b})$ but $Q^{(n)}(\boldsymbol{\nu}|\boldsymbol{b})$ that contains the exact expression of the prior for the n^{th} flux. In fact, at each step we determine the parameters a_n and b_n minimizing $D_{KL}\left[Q^{(n)} \parallel Q\right]$.

3.5.3 EP free energy functional

As we have seen in section 3.5.2 EP algorithm is not a standard variational method, but still we can associate a free energy functional of EP approximation that, as for BP algorithm, can be viewed as an approximation of the exact one. Fixed point of EP algorithm corresponds to the stationary point of the free energy functional derived in appendix E. We report here the final result

$$F_{EP} = -\log Z_Q - \sum_n \log Z_n \tag{3.5.17}$$

where

$$Z_Q = \int d^N\boldsymbol{x} P(\boldsymbol{y}|\boldsymbol{x}) \prod_n \phi_n(x_n) \tag{3.5.18}$$

$$Z_n = \frac{1}{Z_Q}\int d^N\boldsymbol{x} P(\boldsymbol{y}|\boldsymbol{x})\psi_n(x_n) \prod_{m\neq n} \phi_m(x_m) \tag{3.5.19}$$

3.6 Other techniques

In this section we briefly outline several tools used to solve the problems presented in chapter 2. For most of the cases, these approaches will be used to compare the results of

our implementation of the BP, MS or EP algorithms applied to the problems in interest. More precisely, we will present:

- Linear Programming methods. These approaches are used to solve the PCStP, the CS problem (with the L_1 and L_2 regularizations) and to determine the space of solutions of metabolic fluxes in particular cases. The last application falls under the name of Flux Balance Analysis. We will also exploit this technique to solve a maximum weighted matching problem in chapter 5 arising from the update equations of Max Sum.

- Monte Carlo methods. We will briefly explain the Hit-and-Run Monte Carlo algorithm able to sample the space of configurations of metabolic fluxes.

3.6.1 Linear Programming

In this section we will briefly outline the Linear Programming (LP) method to solve optimization problems [46, 47]. A linear program consists in finding a vector $\boldsymbol{x}^*$ among all possible vectors $\boldsymbol{x} \in \mathbb{R}^N$ able to maximize (or minimize) an objecting function; as the name suggests, vectors $\boldsymbol{x}$ must satisfy some linear constraints. Mathematically speaking, we can express this problem as

$$\begin{cases} \max_{\boldsymbol{x}} & \boldsymbol{c}^T\boldsymbol{x} : \\ & \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{cases} \tag{3.6.1}$$

where $\boldsymbol{c} \in \mathbb{R}^N$ is the vector of coefficients of the objective function, $\mathbf{A}$ is a $M \times N$ matrix formalizing the M linear constrains and $\boldsymbol{b} \in \mathbb{R}^M$ is the vector of constant terms. Geometrically, the inequalities constraints define a convex polytope over which the objective function must be maximized. Each vector $\boldsymbol{x}$ satisfying $\mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}$ is a feasible solution of the problem but only $\boldsymbol{x}^*$is an optimal, if unique, solution. However, under certain conditions, there exist infinite solutions or no solutions, if all the possible $\boldsymbol{x}$ are unfeasible.

According to the duality theorem, the optimal value of the objective function is bounded "from above" by the inequality

$$\boldsymbol{c}^T\boldsymbol{x} \leq \boldsymbol{b}^T\boldsymbol{y} \tag{3.6.2}$$

where the vector $\boldsymbol{y}$ is the solution of the *dual* linear program

$$\begin{cases} \min_{\boldsymbol{y}} & \boldsymbol{b}^T\boldsymbol{y}: \\ & \mathbf{A}^T\boldsymbol{y} \geq \boldsymbol{c} \\ & \boldsymbol{y} \geq 0 \end{cases} \tag{3.6.3}$$

If both linear program and its dual linear program have feasible solutions, then the solution is unique and the inequality in (3.6.2) satisfies the equality as:

$$\boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{b}^T\boldsymbol{y}^* \tag{3.6.4}$$

The computing time needed to solve a LP instance is polynomial.

In many practical problems one seeks integer solutions, that is vectors $\boldsymbol{x} \in \mathbb{Z}^N$, but unfortunately the integer linear programming formulation has been proved to belong to the NP-hard class of problems. In practice, one can relax the problem and solve it for real values of the unknowns.

3.6.1.1 Linear programming formulation of the PCStP

Often the problem of minimizing the objective function in (2.1.1) is expressed as a constrained minimization problem over binary auxiliary variables; this mapping leads to an integer linear program.

Due to its infeasibility, one relaxes the integer constraints and seeks a solution of the linear program. There is not a unique mapping; we mention the works in [48, 17, 13] for the interested readers.

As we have seen in (3.6.1), one often defines a *dual* formulation associated with the linear minimization problem that consists, in this case, in a maximization problem. If it can be solved to optimality, then also the corresponding minimization has a unique solution. Unfortunately it is not always the case and often heuristics are used to solve the dual program. The Goemans-Williamson heuristics is in fact an algorithm that tries to solve a dual linear program associated with the PCStP. In the following, we briefly present the Goemans-Williamson (GW) heuristics [19, 49] as described in [50].

3.6.1.1.1 Goemans-Williamson heuristics Let us consider a graph $\mathcal{G}(V, E)$ in which we aim at solving the PCStP. The GW heuristics consists essentially in two steps: the growing and the pruning stages. In the first one nodes are partitioned into disjoint clusters. With each partition C we associate a "moat" variable $y_C \geq 0$ that can be

interpreted as a variable of the dual problem. Clusters are said to be "active" if its moat variable grows after one iteration of the algorithm and "non-active" otherwise. Clusters are merged and deactivated until one active cluster remains. Moreover, edges of the graph are "active" if both ends belong to (different) active clusters, "semi-active" if only one end belongs to an active cluster and "non-active" if ends are members of deactivated clusters or of the same active cluster. We define an auxiliary edges set F which is returned at the end of the process.

At the beginning of the process each node is a singleton active cluster with zero moat variable and $F = \varnothing$. We increase the moat variables associated with active clusters until one of these possible events occurs:

- An "edge constraint" is tight. This means that there is an active (or semi-active) edge $e = (i, j)$ such that $y_{C_i} + y_{C_j} = w_{ij}$ where C_i and C_j are the clusters containing nodes i and j respectively. We merge the two sets in a way that the "new" active cluster contains the history of their moat variables. Thus we add nodes of cluster C_j to C_i and we increment $y_{C_i} \leftarrow y_{C_i} + y_{C_j}$. We keep in memory C_i and we erase C_j decreasing the number of active clusters. Edge e is deactivated and added to F.

- A "cluster constraint" is tight, meaning that there exists an active cluster C such that $y_C = \sum_{i \in C} c_i$. In this case C is deactivated and the number of active clusters is decreased.

The growing step can be simplified introducing the notion of time. Consider the first iteration of the algorithm: all edges are active and all clusters contain only one node. Since the dual variables associated with each cluster C increase at the same rate, we face a tight edge constraint if there exists $(i, j) : y_{C_i} + y_{C_j} = 2y_C = w_{ij}$ or a tight cluster constraint if there exists $C_i : y_C = c_i$. Thus we determine the first event as the one that will occur at time $t = \min(t_e, t_C)$ where $t_e = \min_{(i,j)} \frac{w_{ij}}{2}$ and $t_C = \min_{i \in V} c_i$. After that, we pretend of resetting the time and we apply again the strategy of the first move to the updated scenario taking into account the appearance of semi-active edges, non-active clusters and/or merged clusters. The second event will then occur at time $t = \min\{t_A, t_{SA}, t_C\}$ in which $t_A = \min_{active\,(i,j)} \frac{w_{ij}}{2}$, $t_{SA} = \min_{semi-active\,(i,j)} w_{ij}$ and $t_C = \min_{active\,C} \sum_{i \in C} c_i$. Notice that t_{SA} differs from t_A since one end of a semi-active edge belongs to a non-active cluster. We iterate this procedure until we end up with one active cluster and a subset of edges F forming a tree.

The second step, called "pruning stage", consists in pruning the tree build from F in order to lower the objective function computed in F without affecting the connection of nodes.

We will show 3.6.1.1.1 how to re-weight the weights and the prizes to a graph in order to significantly improve the performances of the Goemans-Williamson heuristics.

3.6.1.2 Convex optimization for CS problem

The priors introduced in section 2.2.1.2 allow us to solve the problem of maximizing the posterior distribution of the CS problem, as a minimization problem of an objective function in a convex space. Let us express the posteriors corresponding to the priors (2.2.8) and (2.2.9) as

$$P_{L_1}(\boldsymbol{x}|\boldsymbol{y}) \propto e^{-\frac{1}{2\Delta}\|\boldsymbol{y}-\mathbf{F}\boldsymbol{x}\|_2} e^{-\beta_1\|\boldsymbol{x}\|_1} \tag{3.6.5}$$

$$P_{L_2}(\boldsymbol{x}|\boldsymbol{y}) \propto e^{-\frac{1}{2\Delta}\|\boldsymbol{y}-\mathbf{F}\boldsymbol{x}\|_2} e^{-\beta_2\|\boldsymbol{x}\|_2} \tag{3.6.6}$$

The MAP approach in section 3.1 is equivalent to solve the following two linear programs

$$\begin{cases} \min_{\boldsymbol{x}} & \frac{1}{2\Delta}\|\boldsymbol{y}-\mathbf{F}\boldsymbol{x}\|_2 + \beta_1\|\boldsymbol{x}\|_1 : \\ & \boldsymbol{x} \in \mathbb{R} \end{cases} \tag{3.6.7}$$

$$\begin{cases} \min_{\boldsymbol{x}} & \frac{1}{2\Delta}\|\boldsymbol{y}-\mathbf{F}\boldsymbol{x}\|_2 + \beta_2\|\boldsymbol{x}\|_2 : \\ & \boldsymbol{x} \in \mathbb{R} \end{cases} \tag{3.6.8}$$

using L_1 or L_2 regularizations respectively. The critical values of the measurements rates α_c are plotted as a function of the sparsity ρ of the signal in figure 2.2.1. Perfect reconstruction is guaranteed for those values of $\alpha < \alpha_c$. We will show in chapter 6 how to achieve perfect reconstruction under the L_1 line (red line in figure 2.2.1) using the EP algorithm.

3.6.1.3 Flux Balance Analysis

One of the most used technique for having access to the solution of (2.2.12)(2.2.11) is the Flux Balance Analysis (FBA) [51–53]. FBA adds to the system of equations in (2.2.12)(2.2.11) the maximization constraint of the growth rate of the organism. Thus it provides the configuration(s) of fluxes such that the biomass reaction has its maximum allowed flux. Mathematically, we can express constraints (2.2.12)(2.2.11) along with the

biomass (BM) maximization as a linear program:

$$\begin{cases} \max_{\nu_{BM}} & \sum_{i=1}^{M} S_{i,BM}\nu_{BM} : \\ & \mathbf{S}\boldsymbol{\nu} = \boldsymbol{b} \\ & \boldsymbol{\nu}^{min} \leq \boldsymbol{\nu} \leq \boldsymbol{\nu}^{max} \end{cases} \tag{3.6.9}$$

The solution is often unique and it corresponds to a point of the polytope. Many experimental works confirm the prediction made by FBA, for instance the growth rate of *in silico* Escherichia Coli [54], but its accuracy is debated under more generic environmental conditions [55]. This tool will be used in chapter 7 to determine the lower bound of the exchange glucose flux for a given growth rate.

3.6.2 Monte Carlo Methods

The Monte Carlo (MC) methods are a family of sampling techniques deeply used in probability theory, statistical physics and combinatorial optimization[2, 1]. Let us suppose that we want to sample a probability distribution $P(\boldsymbol{x})$. Briefly, a Monte Carlo technique consists in constructing a Markov chain that, starting from a random state $\boldsymbol{x}_0$ and using predefined transition rules among configurations, converges to the desired probability $P(\boldsymbol{x})$. The sampling will be as good as the time spent in exploring the configurations is large. Here we will focus on a specific implementation of MC that is called Hit-and-Run.

3.6.2.1 Hit-and-run Monte Carlo for constrained metabolic fluxes

Hit-and-Run is the most effective Monte Carlo sampling method used to sample convex polytopes [56–58]. Here we briefly introduce the main steps of the algorithm applied to the space of solution of metabolic fluxes.

At each iteration t we start from a point $\boldsymbol{\nu}^t = (\nu_1, \ldots, \nu_N)$ within the polytope and we choose a random direction $\boldsymbol{\theta}^t$ and a step size $\lambda \in \mathbb{R}$ such that the point $\boldsymbol{\nu}^t + \lambda\boldsymbol{\theta}^t$ lies on the boundary of the polytope. We then collect a predefined number of points between $\boldsymbol{\nu}^t$ and $\boldsymbol{\nu}^{t+1} = \boldsymbol{\nu}^t + \lambda\boldsymbol{\theta}^t$ and we repeat again these steps starting from the new points until we reach the desired number of explored configurations. To guarantee that the first point lies within the solution space, one often uses linear programming techniques. This method guarantees an estimate of the marginal probability distribution of the fluxes that can be much more informative with respect to FBA outcomes. Nonetheless, it suffers from several bottlenecks.

It is clear that performances of this algorithm are as accurate as we densely sample the polytope, being exact only on the asymptotic limit as standard Monte Carlo algorithms; thus to obtain a reliable sample we need to collect as many points as we can, affecting the overall computing time of the algorithm. It has been proved that this special implementation of the Monte Carlo algorithm converges in polynomial time and the number of steps required to reach convergence is bounded. To obtain a small error ε between the true distribution and the approximated one, the minimum number of steps scales logarithmically with ε but quadratically with respect to the system size times a non-negligible factor 10^{10}[59]. This method is also very sensitive to narrow angles in high dimensions as it can spend a lot of time escaping from them if the directions of the sampling are randomly chosen. A lot effort has been done to improve this strategy trying to solve the issues reported above. We mention for instance the works in [60].

In chapter 7 we will deeply use the HR method to compare our estimate concerning the marginal probability distributions of metabolic fluxes.

Part II

Main contributions

Chapter 4

Solving the PCStP on real-world instances

As explained in section 1.2.2, a combinatorial optimization problem can be mapped into a statistical mechanics problem, where the objective function plays the role of an energy of interacting variables. This is what we are going to derive in the case of the PCStP defined in section 2.1.1. Our goal is to find the ground state of the corresponding statistical mechanics model such that it solves the PCStP. Notice that in addition to the minimization constraint, we have to impose that the solution of the problem is topologically a tree. This global connectivity constraint can be expressed as a set of local constraints over proper variables of a factor graph over which we apply the Max Sum algorithm presented in section 3.3.4.

Our formalism relies on a set of "depth" variables associated with edges that are closely related to the node-based "pointer-variable" described in [21, 20] and briefly mentioned in example 6. At difference to the "pointers" formalism, edge-based variables allow a different formulation of the constraints, called the flat model (see section §4.3), along with the application of modified heuristics (as explained in section §4.4), such as a Max-Sum guided version of the Goemans-Williamson heuristics introduced in 3.6.1.1.1. These improvements are able to provide good solutions for real-world instances (such as grid graphs) where Max Sum alone does not even converge. Part of the developments proposed in the following sections participated to the 11^{th} DIMACS implementation challenge on Steiner Tree Problem, which consisted in a competitive comparison of different techniques on a set of instances chosen to be particularly hard to solve. Our performances maintained a small gap (i.e. were just slightly worse) to the best bound in most cases and obtained best results on several instances of the PCStP.

All the discussion presented in this section is part of the work in [61].

4.1 Factor graph representation

Let us associate with the original graph $\mathcal{G}(V, E)$ a factor graph having the same topology of the original one. Consider a feasible solution T of the PCStP where the distance from the root r and any other leaves can be covered in D hops. Within the solution, we will define for each edge $(i, j) \in E$ an integer variable $d_{ij} \in \{-D, \ldots, D\}$ that is the distance (in hops), or *depth*, from the farthest between i and j to r along the tree. For edges such that both $(i, j), (j, i) \notin E_T$, we conventionally set $d_{ij} = 0$. The sign of non-zero depths will define the orientation of each edge $(i, j) \in E_T$ in the tree with respect to the root r: if $d_{ij} > 0$ node i is "pointing" to the root, that is if one make a step towards j will be closer to r. It is clear that in order to have a consistent representation, the vector $\boldsymbol{d} = \{d_{ij} : (i, j) \in E\}$ so defined satisfies an anti-symmetric condition $d_{ij} = -d_{ji}$ for each $(i, j) \in E$.

To ensure that the assignment of the $\boldsymbol{d}$ variables corresponds to a tree, we need to impose rigid constraints on this vector besides the anti-symmetric condition. As mentioned in the introduction, the global connectivity constraint can be splitted in a family of local constraints on sub-vectors of variables $\boldsymbol{d}_i = \{d_{ji} : j \in \partial i\}$ incident on node i. For each node i we define a proper compatibility function $\psi_i(\boldsymbol{d}_i)$ to be intended as a factor node of the factor graph. As the sign of d_{ij} represents the orientation of edges along the tree, two mutually excluding situations can occur in the neighbor of i. Either there exists exactly one neighbor $j \in \partial i$ such that $d_{ij} > 0$, and for the remaining neighbors $k \in \partial i \setminus j$, either $d_{ki} = 0$ or $d_{ki} = d_{ij} + 1$, or node i does not belong to T, and so $d_{ij} = 0$ for each $j \in \partial i$. The root node r is special, as there is no neighbor closer to r than itself and thus for each neighbor $j \in \partial r$, d_{jr} is either 0 or 1. Formally, allowed configurations of $\boldsymbol{d}_i$ can be encoded by the nonzero arguments of the following functions:

$$\psi_i(\boldsymbol{d}_i) = \prod_{j \in \partial i} \delta_{d_{ji},0} + \sum_{d>0} \sum_{j \in \partial i} \left[\delta_{d_{ji},-d} \prod_{k \in \partial i \setminus j} (\delta_{d_{ki},d+1} + \delta_{d_{ki},0}) \right] \text{ for } i \neq r \tag{4.1.1}$$

$$\psi_r(\boldsymbol{d}_r) = \prod_{j \in \partial i} \left(\delta_{d_{jr},1} + \delta_{d_{jr},0} \right) \tag{4.1.2}$$

where the function "δ" denotes the Kronecker delta.

This model applies for rooted and bounded instances; when this information is not provided as an input of the problem, we need to determine the best root node and the best value of D. We have proposed several rooting procedures along with a scheme to determine D in [61].

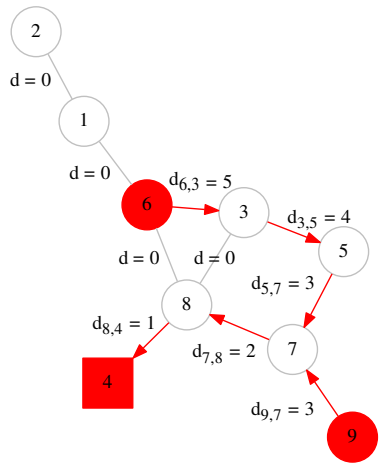


Figure 4.1.1: Solution to the MStP with a proper assignment of the "depth" variables

To give an example of a feasible assignment of these auxiliary variables, we plot in figure 4.1.1 the depth variables associated with the edges of the solution of the MStP in figure 2.1.1.

4.2 Max Sum algorithm

Once the variables nodes and the family of compatibility functions ψ_i are defined, the cost function H in (2.1.1) can be re-written as a function of those variables $\boldsymbol{d}$ representing some tree T, and in such case, $H(T) = H(\boldsymbol{d})$:

$$H(\boldsymbol{d}) = \begin{cases} \infty & \text{if } \prod_i \psi_i(\boldsymbol{d}_i) = 0 \\ \sum_i \left\{ c_i \mathbb{I}[\boldsymbol{d}_i \equiv \mathbf{0}] + \sum_{j \in \partial i} w_{ij} \mathbb{I}[d_{ij} > 0] \right\} & \text{if } \prod_i \psi_i(\boldsymbol{d}_i) = 1 \end{cases} \tag{4.2.1}$$

Thus the original optimization problem is equivalent to seek the optimal configuration

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} H(\boldsymbol{d})$$

As we mentioned in section 1.2.2 we can define a Boltzmann distribution as

$$P_\beta(\boldsymbol{d}) = \frac{1}{Z_\beta} e^{-\beta H(\boldsymbol{d})} \tag{4.2.2}$$

and eventually we investigate the $\beta \to +\infty$. In particular we are interesting in assigning, for each edge $(i,j) \in E$ the optimal state

$$d_{ij}^* = \arg\max_{d_{ij}} P\left(d_{ij}\right) \tag{4.2.3}$$

where $P\left(d_{ij}\right) \propto \sum_{\boldsymbol{d}\setminus d_{ij}} P_\beta\left(\boldsymbol{d}\right)$ is the marginal probability distribution of variables d_{ij}. Unfortunately computing these marginals is as hard as the original problem and one has to approach this computation using some approximation schemes. As described in [61] we apply the Max Sum algorithm on the factor graph just presented, whose update equations can easily derived following the explanations in section 3.3.4; they read

$$h_{ij}\left(d_{ij}\right) \propto \max_{\substack{\boldsymbol{d}_i\setminus d_{ij}: \\ \psi_i(\boldsymbol{d}_i)=1}} \left\{ -c_i \mathbb{I}\left[\boldsymbol{d}_i \equiv \mathbf{0}\right] - \sum_{k\in\partial i} w_{ki} \mathbb{I}\left[d_{ki} > 0\right] + \sum_{k\in\partial i\setminus j} h_{ki}\left(d_{ki}\right) \right\} \quad \text{if } i \neq r \tag{4.2.4}$$

$$h_{rj}\left(d_{rj}\right) \propto \begin{cases} \sum_{k\in\partial r\setminus j} \max\left\{h_{kr}\left(1\right),\, h_{kr}\left(0\right)\right\} & \text{for } d_{rj} = -1, 0 \\ -\infty & \text{for } d_{rj} \neq -1, 0 \end{cases} \tag{4.2.5}$$

where the proportionality signs underline that the normalization condition $\max_{d_{ij}} h_{ij}\left(d_{ij}\right) = 0$ is missing. One iterates the equations in (4.2.4) and (4.2.5) until a numeric convergence is reached. At convergence one can estimate the MS "beliefs" in the zero-temperature limit as

$$M_{ij}^{\beta\to+\infty}\left(d_{ij}\right) \propto h_{ij}\left(d_{ij}\right) + h_{ji}\left(-d_{ij}\right) \tag{4.2.6}$$

and finally determine the so called *decisional* variables

$$d_{ij}^* = \arg\max M_{ij}^{\beta\to+\infty}\left(d_{ij}\right) \tag{4.2.7}$$

that maximize $M_{ij}^{\beta\to+\infty}\left(d_{ij}\right)$. At convergence, the vector $\boldsymbol{d}^*$ so defined constitutes a tree on the original graph, in the sense that the inverse mapping $\boldsymbol{d}^* \mapsto (V_{\boldsymbol{d}^*}, E_{\boldsymbol{d}^*})$with $V_{\boldsymbol{d}^*} = \{i \in V : \exists k \in V : d_{ki} \neq 0\}$ and $E_{\boldsymbol{d}^*} = \{(i,j) : d_{ij} \neq 0\}$ represent an oriented tree satisfying all the hard constraints that, hopefully, minimizes the energy function in (4.2.1) (a short proof is reported at the end of the paragraph).

However, equations in (4.2.4) and (4.2.5) very seldom converge for arbitrary graphs. A very powerful procedure that induces the algorithm to reach a fixed point, is to use a

reinforced set of equations that, at iteration $t+1$, read

$$h_{ij}^{t+1}(d_{ij}) = \max_{\substack{\boldsymbol{d}_i \setminus d_{ij}: \\ \psi_i(\boldsymbol{d}_i)=1}} \left\{ -c_i \mathbb{I}[\boldsymbol{d}_i \equiv \mathbf{0}] - \sum_{\substack{k \in \partial i: \\ d_{ki}>0}} w_{ki} + \sum_{k \in \partial i \setminus j} h_{ki}^{t}(d_{ki}) \right\} + \tag{4.2.8}$$

$$+\gamma_t H_{ij}^{t}(d_{ij}) \tag{4.2.9}$$

$$H_{ij}^{t+1}(d_{ij}) = h_{ij}^{t+1}(d_{ij}) + h_{ji}^{t+1}(-d_{ij}) + \gamma_t H_{ij}^{t}(d_{ij}) \tag{4.2.10}$$

From the point of view of the factor graph the reinforcement scheme consists in adding an extra factor to edge-variables that acts as external fields oriented in the direction of the messages in the previous iteration. The perturbation is governed by the parameter γ_t that increases linearly in time, that is $\gamma_t = t\gamma_0$ for $\gamma_0 \sim 10^{-5}$. For further details about this procedure we refer the interest reader to [21, 61].

We mention that the update steps in (4.2.8)(4.2.10) can be performed in $D|E|$ steps.

Proof Consider a generic graphical model of variable nodes $\boldsymbol{x}$ connected through factor nodes $\boldsymbol{a}$. The energy function can be written as $H(\boldsymbol{x}) = \sum_a H_a(\boldsymbol{x}_a)$. The MS equations, at the fixed point, satisfy the following relations:

$$\begin{cases} h_{i\to a}(x_i) &= \sum_{b\in\partial i\setminus a} u_{b\to i}(x_i) - C \\ u_{a\to i}(x_i) &= \max_{\{x_j : j\in\partial a\setminus i\}} \left[\sum_{j\in\partial a\setminus i} h_{j\to a}(x_j) - H_a(\boldsymbol{x}_a)\right] - C \\ M_i(x_i) &= \sum_{a\in\partial i} u_{a\to i}(x_i) - C \end{cases} \tag{4.2.11}$$

where C is an additive constant that imposes the normalization condition. Notice that these are exactly the update equations in section 3.3.4 in an arbitrary factor graph.

If now we insert the first equation of (4.2.11) in the third equation of (4.2.11), we obtain that each marginal $M_i(x_i)$ can be written as

$$M_i(x_i) = h_{i\to a}(x_i) + u_{a\to i}(x_i) - C \quad \forall a \in \partial i \tag{4.2.12}$$

Using the second equation of (4.2.11) we get

$$M_i(x_i) = h_{i\to a}(x_i) + \max_{\{x_j : j\in\partial a\setminus i\}} \left[\sum_{j\in\partial a\setminus i} h_{j\to a}(x_j) - H_a(\boldsymbol{x}_a)\right] - C \quad \forall a \in \partial i \tag{4.2.13}$$

If the maximum of $M_i(x_i)$, namely

$$\max_{x_i} M_i(x_i) = \max_{\boldsymbol{x}_a} \left[\sum_{j \in \partial a} h_{j \to a}(x_j) - H_a(\boldsymbol{x}_a) \right] - C \quad \forall a \in \partial i \tag{4.2.14}$$

is finite means that there exists a set of variables $\boldsymbol{x}_a$ such that the energy is finite and, thus, the hard constraints are satisfied. The decisional variables computed as

$$x_i^* = \arg\max_{x_i} M_i(x_i) \tag{4.2.15}$$

for each variable node, will constitute a feasible solution to the problem. Notice that the maximum must be non-degenerate; to avoid degeneracy, i.e. to avoid different and feasible solutions with equal energy, one can add small "noisy" terms to the energy terms.

4.3 The *flat* model

One of the main disadvantages of the formalism introduced in section §4.1 is that, as the diameter of the sought solution is bounded by a constant D, the model cannot treat variants of the PCStP where the distance between the root and the leaves of the tree can be any. In principle one should take $D = |V|$to ensure that all nodes can be part of the solution but this choice could severely affect the computing time of the MS update equations. Here we propose a slightly different model for which a maximum allowed depth of $D = |K|$, where K is the sub-set of profitable nodes, guarantees the covering of the entire graph.

In this model, called the *flat* model, we allow chains of edges with identical depth d in the two following cases. A chain can be built between two nodes, let us call them v_0 and v_k if (i) none of the nodes within the chain are profitable vertices and (ii) any node has degree exactly two within the tree. These two conditions ensure that (optimal) configurations satisfying this relaxed set of constraints represent trees; extra cycles with identical depth, containing no terminal, can of course be present, but are sub-optimal in terms of cost. Formally, we would use, instead of the compatibility function in (4.1.1), a function $\psi_i'(\boldsymbol{d}_i)$ defined as:

$$\psi_i'(\boldsymbol{d}_i) = \psi_i(\boldsymbol{d}_i) + \psi_i^{flat}(\boldsymbol{d}_i) \tag{4.3.1}$$

$$\psi_i^{flat}(\boldsymbol{d}_i) = \delta_{c_i,0} \sum_{d>0} \sum_{k \in \partial i} \sum_{l \in \partial i \setminus k} \delta_{d_{ki},-d} \delta_{d_{li},d} \prod_{j \in \partial i \setminus k,l} \delta_{d_{ij},0} \tag{4.3.2}$$

The update equations in the Max Sum formalism have to encompass also this case, and thus

$$h_{ij}(d_{ij}) = \max_{\substack{\boldsymbol{d}_i \setminus d_{ij}: \\ \psi'_i(\boldsymbol{d}_i)=1}} \Big\{ -c_i \mathbb{I}[\boldsymbol{d}_i \equiv \mathbf{0}] - \sum_{k \in \partial i} w_{ki} \mathbb{I}[d_{ki} > 0] + \sum_{k \in \partial i \setminus j} h_{ki}(d_{ki}) \Big\} \tag{4.3.3}$$

$$= \max \Big\{ M_{ij}(d_{ij}), M_{ij}^{flat}(d_{ij}) \Big\} \tag{4.3.4}$$

where $M_{ij}(d_{ij})$ is exactly what we would obtained in the previous model, that is (4.2.4), and $M_{ij}^{flat}(d_{ij})$ is derived from the compatibility function $\psi_i^{flat}(\boldsymbol{d}_i)$

$$M_{ij}^{flat}(d_{ij}) = -w_{ij} \mathbb{I}[d_{ij} > 0] + \max_{\substack{\boldsymbol{d}_i \setminus d_{ij}: \\ \psi_i^{flat}(\boldsymbol{d}_i)=1}} \sum_{k \in \partial i \setminus j} \Big\{ h_{ki}^t(d_{ki}) - w_{ki} \mathbb{I}[d_{ki} > 0] \Big\} \tag{4.3.5}$$

To give an example of the benefits carried by the *flat* representation, consider a solution for the MStP on a two-dimensional squared lattice very few terminals. An example is reported in figure 4.3.1: terminal nodes are red circled while red numbers indicate the "depth"-variables associated with incoming edges of the solution. As we can notice from the plot on the top, proceeding from the root node (here the "39" node) to any of the leaves of the tree, we must increasing the distance for each "step" we perform. The minimum value of the D parameter is 10. Differently, we can exploit the *flat* representation and thus allow very long "chains" with the same depth as appears in the path connecting terminal "13" to the root. In this case, the minimum value for D is 2.

4.4 Max Sum guided heuristics

At difference to local search algorithms for the PCStP [13, 18], MS does not provide any trial solution as the decisional variables defined in (4.2.7), computed before convergence, are in a state of inconsistency, meaning that the hard topological constraints may not be fully satisfied. Here we present how to apply fast heuristics to a "re-weighted" graph in order to obtain feasible solutions for the PCStP after few iterations of the algorithm. Notice that in this case solutions can be sub-optimal in terms of cost but they have the advantageous of resulting unbounded as now the diameter of the solution is not anymore bounded by the parameter D.

The design of the re-weighting ensures that weights and prizes associated with edges and nodes of the original graphs contain information carried by MS algorithm during convergence. More precisely, we apply the following three algorithm:

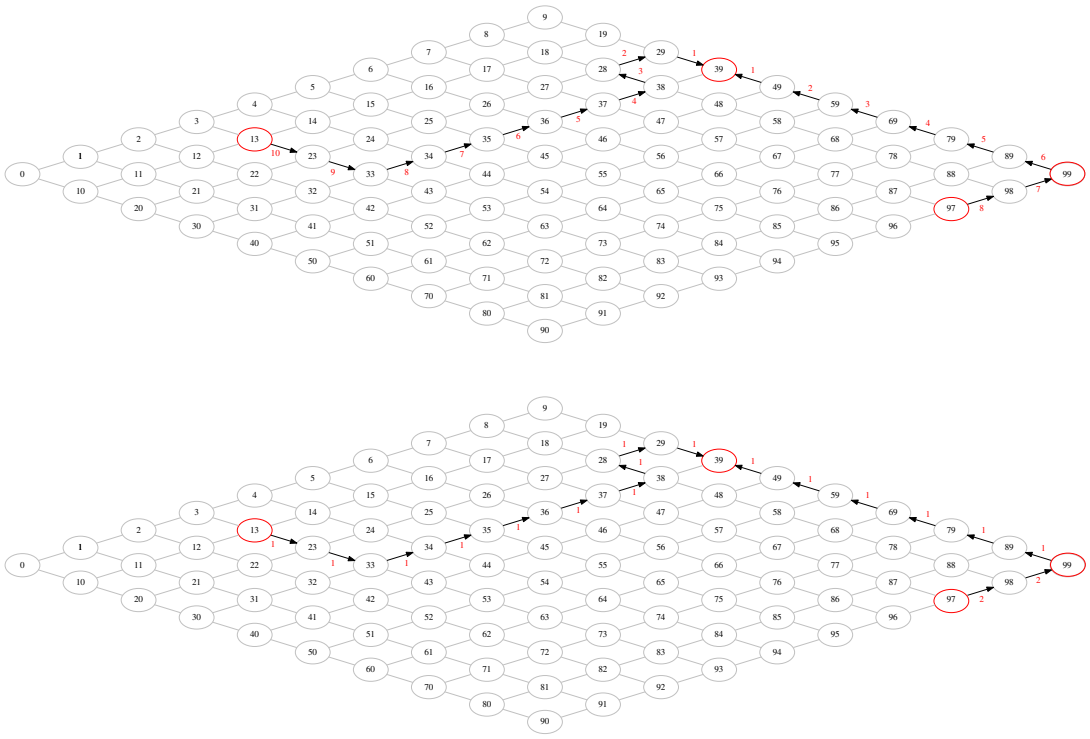


Figure 4.3.1: Solution for the MStP on a lattice with few terminals. Top plot: solution in the original formalism representation. Bottom plot: solution in the *flat* representation

- Minimum Spanning Tree and Shortest Path Tree

  First, we seek a spanning tree $T_H(V_H, E_H)$ (using either Prim or Dijkstra's algorithm for the Minimum Spanning Tree (MST) or the Shortest Path Tree (SPT) respectively) within a graph of temporary weights $\left\{w_{ij}^t : (i,j) \in E\right\}$and temporary prizes $\{c_i^t : i \in V\}$; then we prune it to reduce extra-cost carried by edges that are not responsible for "important" connections within the tree. More precisely, we start from each leaf node $i \in V_H$ with $\partial i = \{j\}$ and we check whether $w_{ij} > c_i$. In this case adding node i to the solution is energetically unfavorable and we delete i and (i,j) from T_H. We recursively repeat this procedure until no such leaf is found. Weights w_{ij}^t will be computed in two ways:

  1. *Re-weighting edges.* A first way sets $w_{ij}^t = \max_{d\neq 0} \left|H_{ij}^t(d)\right|$. This quantity will be strictly positive if the decisional variable $d_{ij}^* = 0$ and will be zero if $d_{ij}^* \neq 0$.

  2. *Re-weighting nodes.* A second way considers the presence of each vertex i in the solution. From the equations, a decisional variable can be assigned to the presence of node i at depth $d \geq 0$ by setting

  $$\begin{aligned} h_i(d) &= \max_{k\in\partial i}\left\{h_{ik}^t(-d) + \sum_{l\in\partial i\setminus k} \max\left\{h_{li}^t(d+1), h_{li}^t(0)\right\}\right\} \quad \text{for } d>0 \\ h_i(0) &= \sum_{k\in\partial i} h_{ki}^t(0) - c_i \end{aligned} \tag{4.4.1}$$

  We will thus force the presence of nodes i such that $\max_{d>0} h_i(d) > h_i(0)$, by adding a large prize C to edges connecting nodes not satisfying this property.

- Goemans-Williamson heuristics

  Before applying the algorithm presented in 3.6.1.1.1 we modify prizes and weights in the following way. For each node we compute $h_i = \max_d h_i(d) - h_i(0)$ defined in (4.4.1). If $h_i > 0$, node i is considered in the intermediate solution and so we increase the prize c_i of a large constant C; otherwise it keeps its original prize. In this way we favor those clusters containing nodes with zero original prize but predicted by MS as Steiner nodes. Edges connecting nodes $i:\ h_i < 0$ have a penalty equal to C so that clusters whose members are not included in MS solution are penalized.

4.5 Results

In this section we report some of the results shown in [61] as an example of the goodness of the algorithms presented in this chapter. To quantitatively compare the different developments, we compute a gap

$$Gap(x,\, y) = \frac{E_x - E_y}{E_y} \cdot 100 \tag{4.5.1}$$

where E_x and E_y are the energies of solutions of implementation x and y respectively. Experiments are labelled depending on which model, heuristics and assignment of weights and/or prizes have been used. All the features of the final algorithm correspond to the following labels:

- "O": this is the original version of the algorithm which competed in the DIMACS challenge. It consists in the Max-Sum algorithm for the normal model joined to the MST; weights are modified as described in *Re-weighting edges* on the facing page.
- "N": we implement the MST heuristics in which weights are computed according to *Re-weighting nodes* on the preceding page.
- "J": here we use the SPT heuristics and weights are modified as in *Re-weighting edges* on the facing page.
- "W": the heuristics is the GW.
- "F": we use the *flat* model. If no additional labels are included, we refer to the MST heuristics with modified weights as in *Re-weighting edges* on the preceding page.

4.5.1 Max Sum against heuristics

To give an example of the benefits carried by the MS-guided heuristics we have applied MS algorithm and all the modified heuristics to a set of grid graphs of size 10x10x10 containing $\{10, 110, 210, 310, 410\}$terminals where we solved the MStP. In fact, many applications of the PCStP arising from electronics (like the VLSI [62]) where the design of circuits can be mapped into variants of the MStP. Pins on a chip can be modelled as nodes on a grid graph connected through several wires, the edges of the graph. For this reason it is of utmost importance solving the PCStP on this kind of graphs. However, being grids very loopy, long correlations among variables nodes of the factor graph

Terminals	"O" energy	MS energy	MS	conv.	MST energy
10	10.56	10.80	11/100	4/10	21.82
110	56.24	61.55	6/100	2/10	77.04
210	81.76	83.13	17/100	4/10	100.93
310	103.62	103.49	25/100	6/10	120.87
410	123.44	124.10	26/100	7/10	137.23

Table 4.1: Average energies for "O", MS, MST and MS convergences for grid graphs 10x10x10 as a function of the number of terminals.

occur and, therefore, the hypothesis behind the Belief Propagation (and thus Max Sum) approximation do not hold in this case. In fact, MS algorithm (even with in the reinforced version) often does not converge. We show here that, instead, our guided heuristics provide good results for the PCStP and the MStP after few iterations of the main algorithms. Energies are always much more smaller than the ones associated with the solution of the original heuristics.

In table 4.1 we report the energies achieved by "O" algorithm, MS and the MST without the re-weighting scheme introduced in section §4.4. For each graph, having a fixed number of terminals, we apply the MST algorithm to 10 realizations of each graph and we compute the energy as the average value among the realizations. For both "O" and MS we run the algorithms 10 times for each of the 10 realizations with different initial conditions. We pick the best energy among the different initializations and then we average over the 10 instances. The fraction of successes over the 10×10 attempts is reported in the left column of "MS conv". In the right column we count how many times MS converged at least one time over the 10 initializations, and we normalize the number of successes with respect to the number of instances per graph.

We can see that MS barely converges on this family of networks with an average fraction of success of 17/100. The heuristic "O" always provides a solution but sometimes is sub-optimal in terms of energy as we can notice from the comparison with the MS ones. We underline that these energies are far below of the ones of the MST heuristic with original edge weights.

4.5.2 3D grid graphs

Here we show the results obtained by our set of implementations for several instances of 3D grid-graphs. For these simulations, we created several 100x100x2 grid-graphs whose edges have weights distributed uniformly in $[0, 1]$ and whose nodes have prizes (only for the PCStP) in the range $[0, 3]$. This choice of the grid size is very popular in the VLSI chips as we will see in chapter 5. We report in the figures 4.5.1 and 4.5.2 the plots of the

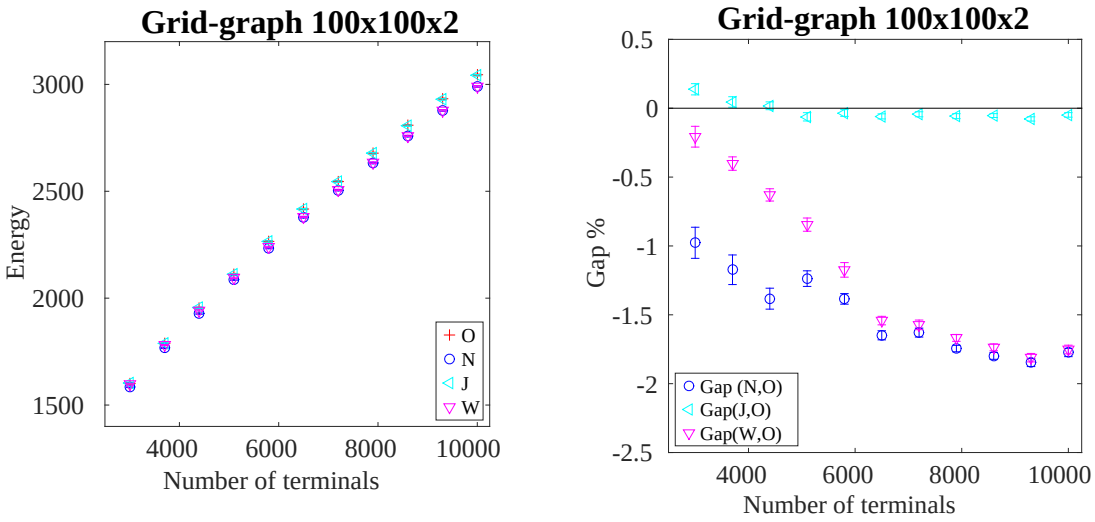


Figure 4.5.1: Left: Energy of the solution for the PCStP for a grid-graph 100x100x2 as a function of the number of terminals. Right: Energy gaps of the "N", "J" and "W" heuristics with respect to "O".

energies (left) and energy gaps (gaps) provided by several algorithms as a function of the number of terminals placed on the graph. In figure 4.5.1 we can appreciate how "N" and "W" outperform the "O" and "J" variants while in figure 4.5.2 the "F" model provides the best energies among all developments.

4.5.3 DIMACS results

In this section we summarize the results obtained by all the implementations of the Max Sum algorithm as with the "normal" or "branching" model in section §4.1 as exploiting the flat model introduced in section §4.3. Instances of the DIMACS competition have been chosen to be particularly challenging as often the optimal solution is not known. A complete set is available here http://dimacs11.cs.princeton.edu/competition.html.

Our implementation "O" participated to the challenge obtaining promising results: energies provided by this implementation had on average a gap of $\pm 0.3\%$ with respect to the best results of the competition. We report in table 4.2 the best improvements of the energy carried by the new implementations in comparison with the "O" algorithm for both MStP and PCStP instances. It is clear that the "N" modified heuristics obtained best performances in the MStP instances while the "O" heuristics combined to the flat model, that is "F", and the modified Goemans-Williamson heuristics "W" significantly improved the results for several instances of the PCStP. All these results correspond

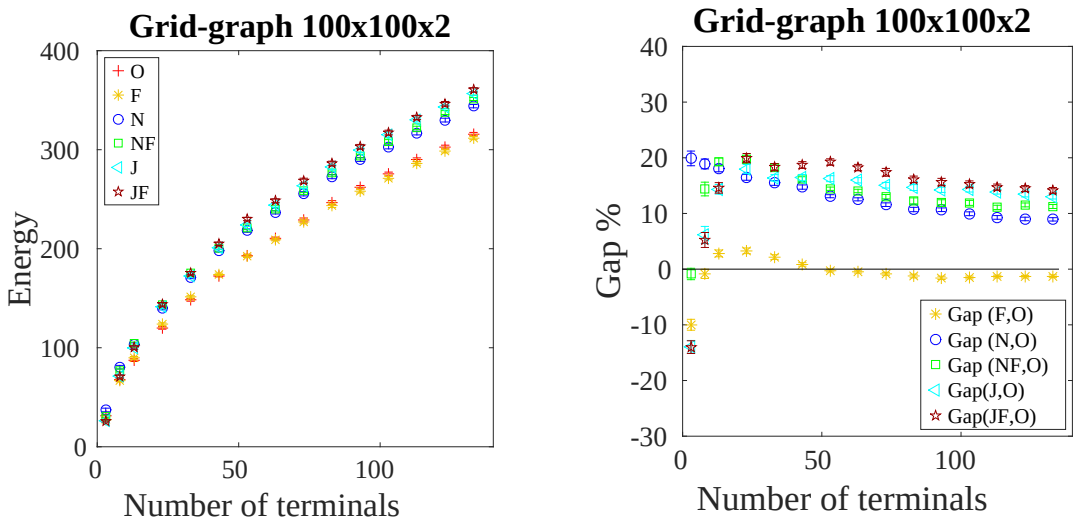


Figure 4.5.2: Left: Energy of the solution for the MStP for a grid-graph 100x100x2 as a function of the number of terminals. Right: Energy gaps of the "F", "N", "NF", "J" and "JF" heuristics with respect to "O".

Instance	**Problem**	**Best energy**	**Algorithm**	**"O" energy**	**Gap %**
world666	MStP	122971	N	130516	-5.78
alut2625	MStP	40183	N	41501	-3.18
es10000fst01	MStP	733237957	N	764631264	-4.11
K400-7	PCStP	485587	W	523885	-7.31
i640-001	PCStP	2932	F J N W	3053	-3.96
i640-221	PCStP	8430	F	8626	-2.27
handsd04	PCStP	525.86	W	584.1	-9.97
handbd13	PCStP	13.23	F J N W	18.1	-26.91
metabol-expr-mice-1	PCStP	11346.93	F J N W	11901.9	-4.66

Table 4.2: Comparison between new implementations and "O" algorithm

to "unbounded" solutions, in the sense that, the parameter D can increase after few iterations of MS algorithm.

We try now to fix $D = |K|$ for the MStP instances of the competition and compare our new implementations for a limited running time of 1200 s. For each heuristic, "N", "O" and "J", we compute the time interval for which the same heuristics combined with the *flat* model, "NF", "F" and "JF", provides a better solution then the normal model. We plot these time thresholds in figure 4.5.3. It is clear that for $es10000fst01$ and $G106ac$ the flat model is more convenient when the running time at our disposal is limited.

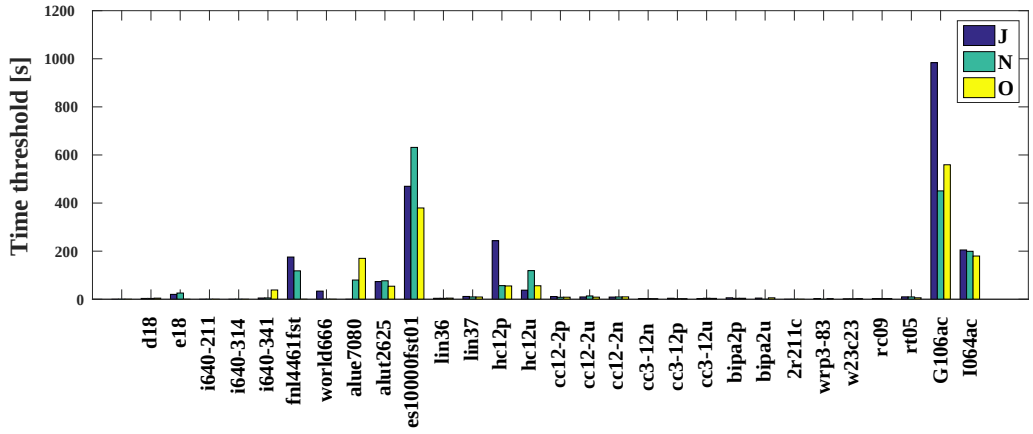


Figure 4.5.3: Time thresholds

4.6 Discussion

We have discussed here several improvements of the message-passing approach to the Steiner Tree Problem on graphs. The first one is the incorporation of heuristics that are able to provide after few iterations of the main algorithm feasible solutions using information coming from the messages before convergence. This is of particular importance because it forces the algorithm to output solutions even in cases in which the tree-like approximation is inaccurate and reinforced Max-Sum equations do not converge. The second one deals with the introduction of an "edge variables" formalism, that allows to deal with a modified *flat* model that removes one limitation of past approaches, namely the need of a large maximum distance D. This results in the "F" variant of the algorithm. Moreover the "edge variables" formulation presented here is also in principle able to accommodate other constraints, such as degree ones. The implementation defined as "O" participated in the 2014 DIMACS Challenge with encouraging results and, as we have shown here, the combinations of other heuristics ("W", "N" , "F" and "J") even improved the energies obtained during the competition.

Chapter 5

Packing of Steiner Trees

In the following we derive three Max-Sum algorithms, one for the V-DStP and two for E-DStP, one more suitable for graphs where the density of terminals is low and the second for instances with low connectivity. Unfortunately cavity equations turn out to be intractable for the E-DStP: to overcome this problem, we introduce two different auxiliary formalisms that allow to compute the equations. The first set of auxiliary variables, called *neighbors occupation variables,* are binary numbers associated with edges of the graph and indicate whether each edge is employed in any of the trees. The algorithm that will follows have to be run over all possible configurations of these binary variables and therefore its computing time results exponential in the (average) degree. In the second case we will show that the computation of the MS update equations is equivalent to solve a maximum matching problem on a small graph. The running time of this second implementation grows exponentially with respect to the number of sub-trees but it is polynomial in the degree. As for the PCStP in chapter 4 we can introduce a reinforced update to facilitate the convergence of the algorithm along with two fast heuristics for the PCStP able to provide fast solutions even in the multiple trees case.

5.1 An arborescence representation

Let us consider a graph $\mathcal{G}(V, E, M)$ as an instance of the V-DStP or the E-DStP and a factor graph, having the same topology of the original graph. Let us define as $\mathcal{G}_\mu$ the tree of the solution for communication μ. As for the PCStP in chapter 4 we introduce a formalism that will allow us to map each variant of the Packing of Steiner trees problem into an optimization problem over proper variables. With each vertex $i \in V$ we associate a factor node obeying constraint ψ_i and with each edge $(i, j) \in E$ a two components variable (d_{ij}, μ_{ij}). Compatibility functions ψ_i are defined in a way that

allowed configurations of variables $(\boldsymbol{d}, \boldsymbol{\mu}) \doteq \{(d_{ij}, \mu_{ij}) : (i,j) \in E\}$ are in one to one correspondence to feasible solutions of the Vertex-disjoint or Edge-disjoint version of the problem. Our choice of the edge-variables is similar to the one adopted in chapter 4 but here, in addition to a "depth" component we introduce a "communication" μ_{ij} by which we label edges forming different trees.

The variable μ_{ij} takes value from the set $\{0, 1, \ldots, M\}$ and denotes to which sub-graph, if any, does the edge belongs; the state $\mu_{ij} = 0$ will conventionally mean that no tree employs the edge (i,j). Components $d_{ij} \in \{-D, \ldots, 0, \ldots, D\}$ have a meaning of "depth" or "distance" within the sub-graph as for the single tree case described in section §4.1. Thus, every edge satisfies $\mu_{ij} = \mu_{ji}$ and the anti-symmetric condition $d_{ij} = -d_{ji}$ must hold. When $d_{ij} = 0$ we conventionally associate $\mu_{ij} = 0$.

In order to ensure Steiner sub-graphs to be trees, i.e. to be connected and acyclic, we impose local constraints on variables $\boldsymbol{d}_i = \{d_{ij} : j \in \partial i\}$ and $\boldsymbol{\mu}_i = \{\mu_{ij} : j \in \partial i\}$. For each node i we aim at defining a proper compatibility function $\psi_i(\boldsymbol{d}_i, \boldsymbol{\mu}_i)$ equals to one if and only if the constraints are *locally* satisfied or zero otherwise. Being the interactions among nodes different as we deal with the V-DStP or the E-DStP, we define two compatibility functions, ψ_i^V and ψ_i^E, for the two problems.

5.1.1 Constraints for the Vertex-Disjoint Steiner trees problem

In the V-DStP a node can belong to none of the solution-trees or at most one sub-graph $\mathcal{G}_\mu$; so, if a vertex i is member of a Steiner tree, all its neighbors can participate to the same communication or being unemployed. A directed tree structure is guaranteed if there exists only one neighbor $j \in \partial i$ such that $\mu_{ij} \neq 0$ and $d_{ij} > 0$, meaning that node j is one step, or "hop", closer than i to the root; all remaining neighbors $k \in \partial i \backslash j$ can either not enter in any solution or be member of tree $\mathcal{G}_{\mu_{ij}}$at the distance $d_{ki} = d_{ij} + 1$ from root $r_{\mu_{ij}}$. We define a compatibility function ψ_i^V, relative to each node i, which takes value one if all topological constraints relative to its neighbors are satisfied and zero otherwise. Being nodes sets of the solution non-overlapping, we can consider separately the belonging of a vertex to a particular tree. Thus, this function can be expressed as the sum over all possible trees of a single-tree compatibility function ψ_i^μ which takes into account the constraints related to the depth components. Its mathematically expression

is reported in (5.1.1).

$$\psi_i^V(\boldsymbol{d}_i, \boldsymbol{\mu}_i) = \sum_{\mu=1}^{M} \psi_i^\mu(\boldsymbol{d}_i, \boldsymbol{\mu}_i) \tag{5.1.1}$$

$$\psi_i^\mu(\boldsymbol{d}_i, \boldsymbol{\mu}_i) = \prod_{j\in\partial i} \delta_{\mu_{ji},0}\delta_{d_{ji},0} + \tag{5.1.2}$$

$$+ \sum_{d>0}\sum_{j\in\partial i}\left[\delta_{\mu,\mu_{ji}}\delta_{d_{ji},-d}\prod_{k\in\partial i\setminus j}\left(\delta_{\mu,\mu_{ki}}\delta_{d_{ki},d+1} + \delta_{\mu_{ki},0}\delta_{d_{ki},0}\right)\right]$$

5.1.2 Constraints for the Edge-Disjoint Steiner trees problem

Differently from the V-DStP, in the E-DStP a vertex can belong either to none or can host more than one communication with the constraint that the local tree structure must be concurrently satisfied for any possible sub-graph. In the first case we must admit configurations in which $\boldsymbol{d}_i = \mathbf{0}$ if $\boldsymbol{\mu}_i = \mathbf{0}$. For the second case, as a subset of neighbors $V_k \subseteq \partial i$ are members of a Steiner tree μ their distances d_{ki} are different from zero if $\mu_{ki} = \mu$ for any $k \in V_k$ and in addition must satisfy the topological constraints. We can mathematically express such conditions through the compatibility function

$$\psi_i^E(\boldsymbol{d}_i, \boldsymbol{\mu}_i) = \prod_{\mu=1}^{M}\left[\prod_{k\in\partial i}\delta_{d_{ki}\delta_{\mu_{ki},\mu},0}\right. \tag{5.1.3}$$

$$\left. + \sum_{d>0}\sum_{k\in\partial i}\delta_{d_{ki}\delta_{\mu_{ki},\mu},-d}\prod_{l\in\partial i\setminus k}\left(\delta_{d_{li}\delta_{\mu_{li},\mu},d+1} + \delta_{d_{li}\delta_{\mu_{li},\mu},0}\right)\right] \tag{5.1.4}$$

Note that if we define $\tilde{d_{ki}} = d_{ki}\delta_{\mu_{ki},\mu}$ we can express again (5.1.3) as

$$\psi_i^E(\mathbf{d}_i, \boldsymbol{\mu}_i) = \prod_{\mu=1}^{M}\left[\prod_{k\in\partial i}\delta_{\tilde{d}_{ki},0} + \sum_{d>0}\sum_{k\in\partial i}\delta_{\tilde{d}_{ki},-d}\prod_{l\in\partial i\setminus k}\left(\delta_{\tilde{d}_{li},d+1} + \delta_{\tilde{d}_{li},0}\right)\right] \tag{5.1.5}$$

$$= \prod_{\mu=1}^{M}\psi_i^\mu\left(\tilde{\boldsymbol{d}}_i, \boldsymbol{\mu}_i\right) \tag{5.1.6}$$

in which the function ψ_i^μ is equivalent to (5.1.2).

5.2 Boltzmann distribution and marginals

The formalism introduced in section §5.1 allows us to map each solution of the packing of Steiner Trees to a certain assignment of variables $\boldsymbol{d} = \{d_{ij} : (i,j) \in E\}$and $\boldsymbol{\mu} = \{\mu_{ij} : (i,j) \in E\}$ of the associated factor graph. The cost function in (2.1.3) can be then

expressed in terms of the new variables satisfying the constraints, that is:

$$H(\boldsymbol{d},\boldsymbol{\mu}) = \begin{cases} \sum_{\mu=1}^{M}\left[\sum_{i\in V} c_i^{\mu}\mathbb{I}\left[\boldsymbol{\mu}_i \neq \boldsymbol{\mu}\right] + \sum_{\substack{d_{ij}>0:\\ \mu_{ij}=\mu}} w_{ij}\right] & if \quad \prod_{i\in V}\psi_i\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right)=1 \\ +\infty & otherwise \end{cases} \tag{5.2.1}$$

where for sake of simplicity we consider the "homogeneous" case $w_{ij}^{\mu} = w_{ij}\ \forall\mu$ and ψ_i can be either equal to ψ_i^V or ψ_i^E. The expression $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}$ means that none of the neighbors $k \in \partial i$ satisfies $\mu_{ki} = \mu$.

The Boltzmann-Gibbs distribution associated with the energy $H(\boldsymbol{d},\boldsymbol{\mu})$ is given by

$$P(\boldsymbol{d},\boldsymbol{\mu}) = \frac{\prod_i \psi_i\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) e^{-\beta H(\boldsymbol{d},\boldsymbol{\mu})}}{Z_\beta} \tag{5.2.2}$$

and the normalization constant reads

$$Z_\beta = \sum_{\boldsymbol{d},\boldsymbol{\mu}}\prod_i \psi_i\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) e^{-\beta H(\boldsymbol{d},\boldsymbol{\mu})}$$

As standard, we investigate the limit $\beta \to +\infty$ where the distribution will be concentrated in the configuration(s) that minimizes $H(\boldsymbol{d},\boldsymbol{\mu})$ that are exactly the solutions of the optimization problems. Thus we are interesting in determining, for each edge $(i,j)\in E$, the assignment of variables that maximizes the marginal probability distribution P_{ij} defined as:

$$P_{ij}\left(\tilde{d}_{ij},\tilde{\mu}_{ij}\right) = \sum_{\boldsymbol{d},\boldsymbol{\mu}} P(\boldsymbol{d},\boldsymbol{\mu})\,\delta_{d_{ij},\tilde{d}_{ij}}\delta_{\mu_{ij},\tilde{\mu}_{ij}} \tag{5.2.3}$$

Unfortunately the computation of (5.2.3) is as intractable as the original problem.

We aim at calculating an approximation of such marginals via BP or, eventually, MS algorithm. At finite β the BP equations on our factor graph are:

$$\begin{cases} m_{ij}\left(d_{ij},\mu_{ij}\right) & = \frac{1}{Z_{ij}}\sum_{\substack{\{d_{ki},\mu_{ki}\}:\\ k\in\partial i\setminus j}} \psi_i\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]}\prod_{k\in\partial i\setminus j} n_{ki}\left(d_{ki},\mu_{ki}\right) \\ n_{ki}\left(d_{ki},\mu_{ki}\right) & = e^{-\beta w_{ki}\mathbb{I}[d_{ki}>0]} m_{ki}\left(d_{ki},\mu_{ki}\right) \end{cases} \tag{5.2.4}$$

where

$$Z_{ij} = \sum_{\{d_{ij},\mu_{ij}\}} m_{ij}\left(d_{ij},\mu_{ij}\right)$$

is the normalization constant or partial partition function. The system of equations in (5.2.4) can be seen as fixed point equations that can be solved iteratively. Starting from a set of initial cavity marginals at time $t = 0$, we iterate the right-hand-side of (5.2.4) until numerical convergence to a fixed point is reached. At this point we estimate marginals in (5.2.3) computing the *cavity fields* defined as:

$$M_{ij}\left(d_{ij}, \mu_{ij}\right) \propto n_{ij}\left(d_{ij}, \mu_{ij}\right) n_{ji}\left(-d_{ij}, \mu_{ij}\right) \tag{5.2.5}$$

The $\beta \to \infty$ limit of the BP equations lead to the Max-Sum algorithm ad the corresponding messages $h_{ij}\left(d_{ij}, \mu_{ij}\right)$ flowing within the factor graph. At convergence we can extract our optimal assignment of variables by the computation of the *decisional* variables

$$\left(d^*_{ij}, \mu^*_{ij}\right) = \arg \max_{(d_{ij}, \mu_{ij})} M^{\beta \to +\infty}_{ij}\left(d_{ij}, \mu_{ij}\right) \tag{5.2.6}$$

$$M^{\beta \to +\infty}_{ij}\left(d_{ij}, \mu_{ij}\right) = h_{ij}\left(d_{ij}, \mu_{ij}\right) + h_{ji}\left(-d_{ij}, \mu_{ij}\right) - C' \tag{5.2.7}$$

where C' is a constant that is added to guarantee that normalization condition in the zero-temperature limit, that is $\max_{(d_{ij}, \mu_{ij})} M^{\beta \to +\infty}_{ij}\left(d_{ij}, \mu_{ij}\right) = 0$, is satisfied. In the following sections we will show how to derive equations for the cavity marginals and cavity fields, for finite β and in the limit $\beta \to +\infty$, depending on we are dealing with the V-DStP or the E-DStP problem. Notice that for these variants it is possible to implement the reinforcement scheme as in section §4.2.

5.3 The cavity equations

5.3.1 Vertex-disjoint Steiner trees Problem

To derive the Belief Propagation equations for the V-DStP problem is sufficient to impose $\psi_i\left(\boldsymbol{d}_i, \boldsymbol{\mu}_i\right) = \psi^V_i\left(\boldsymbol{d}_i, \boldsymbol{\mu}_i\right)$ in (5.2.4). Equations for messages can be easily obtained by using the properties of Kronecker delta functions in $\psi^V_i\left(\boldsymbol{d}_i, \boldsymbol{\mu}_i\right)$ and then, by the change of variables explained in (3.3.4), we will determine a Max-Sum algorithm for this variant.

We can differentiate three cases depending on we are updating messages m_{ij} for positive, negative or null depth d_{ij}:

$$\begin{cases} m_{ij}(d,\mu) = e^{-\beta\sum_{\nu\neq\mu} c_i^\nu} \prod_{k\in\partial i\setminus j} \left[n_{ki}(d+1,\mu) + n_{ki}(0,0)\right] & \forall d>0, \mu\neq 0 \\ m_{ij}(d,\mu) = e^{-\beta\sum_{\nu\neq\mu} c_i^\nu} \sum_{k\in\partial i\setminus j} n_{ki}(d+1,\mu) \times & \\ \qquad\qquad \times \prod_{l\in\partial i\setminus\{j,k\}} \left[n_{li}(d,\mu) + n_{li}(0,0)\right] & \forall d<0, \mu\neq 0 \\ m_{ij}(0,0) = e^{-\beta\sum_{\mu} c_i^\mu} \prod_{k\in\partial i\setminus j} n_{ki}(0,0) + \sum_{\mu\neq 0}\sum_{d<0} m_{ij}(d,\mu) & \end{cases} \tag{5.3.1}$$

Substituting $h_{ij}(d_{ij},\mu_{ij}) = \lim_{\beta\to+\infty} n_{ij}(d_{ij},\mu_{ij})$ in (5.3.1) we obtain the Max-Sum equations:

$$\begin{cases} h_{ij}(d,\mu) = -w_{ij} - \sum_{\nu\neq\mu} c_i^\nu + \sum_{k\in\partial i\setminus j} \max\left\{h_{ki}(d+1,\mu), h_{ki}(0,0)\right\} & \forall d>0, \mu\neq 0 \\ h_{ij}(d,\mu) = -\sum_{\nu\neq\mu} c_i^\nu + \max_{k\in\partial i\setminus j}\left[h_{ki}(d+1,\mu) + w_{ki} + \right. & \\ \qquad\qquad \left. + \sum_{l\in\partial i\setminus\{j,k\}} \max\left\{h_{li}(d,\mu), h_{li}(0,0)\right\}\right] & \forall d<0, \mu\neq 0 \\ h_{ij}(0,0) = \max\left\{-\sum_{\mu} c_i^\mu + \sum_{k\in\partial i\setminus j} h_{ki}(0,0),\ \max_{\mu\neq 0}\max_{d<0} h_{ij}(d,\mu)\right\} & \end{cases} \tag{5.3.2}$$

5.3.2 Edge-disjoint Steiner trees problem

The Belief Propagation equations for the E-DStP are the following:

$$m_{ij}(d_{ij},\mu_{ij}) = \sum_{\substack{\{d_{ki},\mu_{ki}\}:\\ k\in\partial i\setminus j}} \psi_i^E(\boldsymbol{d}_i,\boldsymbol{\mu}_i)\, e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]} \prod_{k\in\partial i\setminus j} n_{ki}(d_{ki},\mu_{ki}) \tag{5.3.3}$$

To compute (5.3.3) we can define:

$$Z_i = \sum_{\boldsymbol{d}_i,\boldsymbol{\mu}_i} \psi_i^E(\boldsymbol{d}_i,\boldsymbol{\mu}_i)\, e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i\neq\boldsymbol{\mu}]} \prod_{k\in\partial i} n_{ki}(d_{ki},\mu_{ki}) \tag{5.3.4}$$

and then calculate messages $m_{ij}(d_{ij},\mu_{ij})$ through (5.3.4) by temporarily setting

$$n_{ji}(d_{ji},\mu_{ji}) = \delta_{-d_{ij},d_{ji}}\delta_{\mu_{ij},\mu_{ji}}$$

Due to the explicit expression of ψ_i^E message-passing equations become intractable and therefore they cannot be efficiently implemented. In the following subsections we propose two different approaches for the computation of (5.3.4) where we make use of two different auxiliary variables. The first algorithm is based on binary occupation variables associated with nodes that scales exponentially in their degree but is linear in M; the second one consists in a mapping between the E-DStP update equations and

the matching problem on a proper bipartite graph, that, in the $\beta \to +\infty$, becomes a maximum matching problem which may be more efficient for vertices with large degrees; however, it scales exponentially in M.

5.3.2.1 Neighbors occupation formalism

Suppose of associating with each vertex $i \in V$ a vector $\boldsymbol{x} = \{0,1\}^{|\partial i|}$. A feasible assignment of these auxiliary variables is guaranteed if, for every link $(i,k) \in E$ incident on i, we will impose $x_k = 1$ if such edge belongs to a tree (i.e. $d_{ki} \neq 0$ and consequently $\mu_{ki} \neq 0$) or $x_k = 0$ otherwise (for $\mu_{ki} = 0, d_{ki} = 0$). Variables $(\boldsymbol{d}_i, \boldsymbol{\mu}_i)$ must locally satisfy the following identity $\prod_{k\in\partial i} \mathbb{I}\left[x_k = 1 - \delta_{d_{ki},0}\right] = 1$ for every node $i \in V$. We introduce this expression in (5.3.4) obtaining:

$$Z_i = \sum_{\boldsymbol{d}_i,\boldsymbol{\mu}_i} \psi_i^E\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i \neq \boldsymbol{\mu}]} \sum_{\boldsymbol{x}} \prod_{j\in\partial i} \mathbb{I}\left[x_j = 1 - \delta_{d_{ji},0}\right] n_{ji}\left(d_{ji},\mu_{ji}\right) \tag{5.3.5}$$

$$= \sum_{\boldsymbol{x}} Z_{\boldsymbol{x}}^M \tag{5.3.6}$$

where $Z_{\boldsymbol{x}}^M$ is defined by taking $q = M$ of the following expression

$$Z_{\boldsymbol{x}}^q \equiv \sum_{\substack{\boldsymbol{d}_i,\boldsymbol{\mu}_i \\ \mu_{ki}\leq q}} \psi_i^E\left(\boldsymbol{d}_i,\boldsymbol{\mu}_i\right) e^{-\beta\sum_\mu c_i^\mu \mathbb{I}[\boldsymbol{\mu}_i \neq \boldsymbol{\mu}]} \prod_{k\in\partial i} \mathbb{I}\left[x_k = 1 - \delta_{d_{ki},0}\right] n_{ki}\left(d_{ki},\mu_{ki}\right) \tag{5.3.7}$$

that can be computed using the following recursion:

$$Z_{\mathbf{x}}^q = \sum_{\boldsymbol{y}\leq\boldsymbol{x}} \Bigg[e^{-\beta c_i^q} \prod_{\substack{k\in\partial i \\ y_k=0 \\ x_k=1}} n_{ki}\left(0,0\right) + \tag{5.3.8}$$

$$+ \sum_{d>0} \sum_{\substack{j\in\partial i \\ y_j=0 \\ x_j=1}} n_{ji}(-d,q) \prod_{\substack{k\in\partial i\setminus j \\ y_k=0 \\ x_k=1}} \left[n_{ki}\left(d+1,q\right) + n_{ki}\left(0,0\right)\right] \Bigg] Z_{\boldsymbol{y}}^{q-1}$$

$$Z_{\mathbf{x}}^0 = e^{-\beta\sum_\mu c_i^\mu} \prod_{j\in\partial i} \delta_{x_j,0} n_{ji}\left(0,0\right) \tag{5.3.9}$$

where with $\boldsymbol{y} \leq \boldsymbol{x}$ we mean all possible vectors $\boldsymbol{y} = \{0,1\}^{|\partial i|}$ satisfying

$$y_k = \begin{cases} y_k \leq x_k & \text{if } \mu_{ki} \neq q \\ 0 & \text{if } \mu_{ki} = q \end{cases} \tag{5.3.10}$$

We do not report here the proof of the equivalence between (5.3.8) and (5.3.7). We can write the expressions above in the Max-Sum formalism. Define $F_i = \lim_{\beta\to+\infty} \frac{1}{\beta} \log Z_i$ and express it as function of Max-Sum messages

$$h_{ij}\left(d_{ij}, \mu_{ij}\right) = \lim_{\beta\to+\infty} \frac{1}{\beta} \log n_{ij}\left(d_{ij}, \mu_{ij}\right) \tag{5.3.11}$$

as

$$F_i = \max_{\substack{\boldsymbol{d}_i, \boldsymbol{\mu}_i \\ \psi_i^E(\boldsymbol{d}_i, \boldsymbol{\mu}_i)=1}} \max_{\boldsymbol{x}} \left[\sum_{k\in\partial i} \log \mathbb{I}\left[x_k = 1 - \delta_{d_{ki},0}\right] + h_{ki}(d_{ki}, \mu_{ki}) - \sum_{\mu} c_i^{\mu} \mathbb{I}\left[\boldsymbol{\mu}_i \neq \boldsymbol{\mu}\right] \right] \tag{5.3.12}$$

where the function $\sum_{k\in\partial i} \log \mathbb{I}\left[x_k = 1 - \delta_{d_{ki},0}\right]$ takes value zero if variables satisfy the constraints or minus infinity otherwise. As in the BP formulation, we rewrite it as:

$$F_i = \max_{\boldsymbol{x}} F_{\boldsymbol{x}}^M \tag{5.3.13}$$

with

$$F_{\boldsymbol{x}}^M = \max_{\substack{\boldsymbol{d}_i, \boldsymbol{\mu}_i \\ \psi_i^E(\boldsymbol{d}_i, \boldsymbol{\mu}_i)=1}} \sum_{k\in\partial i} \left[\log \mathbb{I}\left[x_k = 1 - \delta_{d_{ki},0}\right] + h_{ki}\left(d_{ki}, \mu_{ki}\right) - \sum_{\mu} c_i^{\mu} \mathbb{I}\left[\boldsymbol{\mu}_i \neq \boldsymbol{\mu}\right] \right]$$

It is computed recursively from

$$F_{\boldsymbol{x}}^q = \max_{\boldsymbol{y}\leq\boldsymbol{x}} \left\{ F_{\boldsymbol{y}}^{q-1} + \max\left\{f_0, f_d\right\} \right\} \tag{5.3.14}$$

$$F_{\mathbf{x}}^0 = -\sum_{\mu} c_i^{\mu} + \log \mathbb{I}\left[\mathbf{x} = \mathbf{0}\right] + \sum_{k\in\partial i} h_{ki}\left(0, 0\right) \tag{5.3.15}$$

where

$$f_0 = \sum_{\substack{k\in\partial i \\ y_k=0 \\ x_k=1}} h_{ki}\left(0, 0\right) - c_i^q \tag{5.3.16}$$

$$f_d = \max_{d>0} \left[\max_{\substack{k\in\partial i \\ y_k=0 \\ x_k=1}} h_{ki}\left(-d, q\right) + \sum_{\substack{l\in\partial i\setminus k \\ y_l=0 \\ x_l=1}} \max\left[h_{li}\left(d+1, q\right), h_{li}\left(0, 0\right)\right] \right] \tag{5.3.17}$$

5.3.2.2 Mapping into a weighted matching problem

We will show an alternative method for the computation of the update rules of the messages. Let us introduce an auxiliary vector $\boldsymbol{s} \in \{0, 1, \ldots, D\}^M$ associated with any vertex of the graph. Each component s_μ takes value in the set of the possible positive depths $\{1, \ldots, D\}$ if this node is member of communication μ or 0 otherwise. For a non-root node i member of the communication μ, there exists exactly one neighbor k such that $d_{ik} > 0$, $d_{ki} = -s_{\mu_{ki}}\, \mu_{ki} = \mu$ and for the remaining ones, $d_{li}\delta_{\mu_{li},\mu} = s_{\mu_{ki}} + 1$ or $d_{li}\delta_{\mu_{li},\mu} = 0$, $l \in \partial i \backslash k$. The compatibility function for E-DStP can be expressed as a function of the new variables as:

$$\psi_i^E\left(\boldsymbol{d}_i, \boldsymbol{\mu}_i\right) = \prod_{\mu=1}^{M} \left[\sum_{s_\mu > 0} \sum_{k \in \partial i} \delta_{\tilde{d}_{ki}, -s_\mu} \prod_{l \in \partial i \backslash k} \left(\delta_{\tilde{d}_{li}, s_\mu + 1} + \delta_{\tilde{d}_{li}, 0} \right) + \prod_{k \in \partial i} \delta_{\tilde{d_{ki}}, 0} \right] \tag{5.3.18}$$

Let us define the binary vector $\boldsymbol{t}$ whose component $t_{k\mu}$ takes value 1 if $k \in \partial i$ is member of communication μ and $s_\mu > 0$ or 0 otherwise; it can be proven that Z_i can be computed as

$$Z_i = \sum_{\boldsymbol{s}} R_{\boldsymbol{s}} Z_{\boldsymbol{s}} \tag{5.3.19}$$

where

$$R_{\boldsymbol{s}} = \prod_{k \in \partial i} \left[\sum_{\nu} n_{ki}\left(s_\nu + 1, \nu\right) + n_{ki}\left(0, 0\right) \right] \tag{5.3.20}$$

$$\begin{aligned} Z_{\boldsymbol{s}} = & \sum_{\boldsymbol{t}} \prod_{\mu} \mathbb{I}\left[\sum_{k \in \partial i} t_{k\mu} = 1 - \delta_{s_\mu, 0} \right] \prod_{k \in \partial i} \mathbb{I}\left[\sum_{\mu} t_{k\mu} \leq 1 \right] \times \\ & \times \prod_{k \in \partial i} \prod_{\mu} e^{-\beta c_i^\mu \mathbb{I}[s_\mu = 0]} \left[\frac{n_{ki}\left(-s_\mu, \mu\right)}{\sum_\nu n_{ki}\left(s_\nu + 1, \nu\right) + n_{ki}\left(0, 0\right)} \right]^{t_{k\mu}} \end{aligned} \tag{5.3.21}$$

The term $Z_{\mathbf{s}}$ is the partition function of a matching problem on the complete bipartite graph $G = (V = A \cup B, E = A \times B)$ with $A = \partial i$ and $B = \{\mu : s_\mu > 0\}$ where the energy of a matching is

$$H\left(\boldsymbol{t}\right) = \sum_{k\mu} t_{k\mu} \log \frac{n_{ki}\left(-s_\mu, \mu\right)}{\sum_\nu n_{ki}\left(s_\nu + 1, \mu\right) + n_{ki}\left(0, 0\right)} - c_i^\mu \mathbb{I}\left[s_\mu = 0\right]$$

In general, the partition function $Z_{\mathbf{s}}$ is hard to compute exactly, because it corresponds to the calculation of a matrix *permanent* which is computationally intractable. Fortunately, the situation is much easier in the $\beta \to \infty$ limit: using $h_{ki}\left(-s_\mu, \mu\right) = \frac{1}{\beta} \log n_{ki}\left(-s_\mu, \mu\right)$ and taking the limit $\beta \to \infty$, the computation of $F_i = \frac{1}{\beta} \log Z_i$ reduces

to the evaluation of

$$F_i = \max_{\boldsymbol{s}} \left[\frac{1}{\beta} \left(\log R_{\boldsymbol{s}} + \log Z_{\boldsymbol{s}} \right) \right] \tag{5.3.22}$$

$$= \max_{\boldsymbol{s}} \left\{ \sum_{k \in \partial i} \max \left[\max_{\mu} h_{ki} \left(s_\mu + 1, \mu \right), \; h_{ki} \left(0, 0 \right) \right] + F_{\boldsymbol{s}} \right\} \tag{5.3.23}$$

To evaluate the second term $F_{\boldsymbol{s}} = \frac{1}{\beta} \log Z_{\boldsymbol{s}}$ we need to solve a maximum matching problem on a bipartite graph which can be done in polynomial time. More precisely, for each assignment of the $\boldsymbol{s}$ we can define the weights $w_{k\mu}$ associated with each edge (k, μ) as:

$$w_{k\mu} = \begin{cases} h_{ki} \left(-s_\mu, \mu \right) - \max_\nu \max \left\{ h_{ki} \left(s_\nu + 1, \nu \right), h_{ki} \left(0, 0 \right) \right\} & \text{if } s_\mu > 0 \\ -c_i^\mu & \text{if } s_\mu = 0 \end{cases} \tag{5.3.24}$$

and solve the constrained maximization problem

$$\begin{cases} F_{\boldsymbol{s}} = \max \sum_{(k,\mu)} w_{k\mu} t_{k\mu} & : \\ \sum_{k \in \partial i} t_{k\mu} \leq 1 & \mu = \{1, \ldots, M\} \\ \sum_{\mu} t_{k\mu} \leq 1 & \{k : k \in \partial i\} \end{cases} \tag{5.3.25}$$

Notice that variables $\boldsymbol{t}$ are binary variables and the system in (5.3.25) describe an integer linear program. As discussed in section §3.6.1 one solves the corresponding linear program, that is the same problem but with real variables $\boldsymbol{t}$. This problem is known as the bipartite maximum weighted matching problem whose optimal solution is proven to be integer, i.e. for binary $\boldsymbol{t}$.

5.4 Max Sum for loopy graphs

As we have described in chapter 4, a reinforcement procedure coupled to the use of modified fast heuristics ensure to find good (and unbounded) solutions after few iterations of the algorithm. The same can the implemented for the V-DStP and the E-DStP with the following remark. Heuristics are applied to the graph for any communication and we build the solution for E-DStP or V-DStP as a superposition of single-tree solutions. Notice that heuristics are sequentially applied, i.e. we consider one communication at the time, and depending on we are dealing with V-DStP or E-DStP, edges (and Steiner

nodes for the V-DStP) selected in the first spanning trees cannot be further used by the successive applications.

5.5 Preliminary results

5.5.1 VLSI

We report here some preliminary results for benchmark instances of circuit layout where we solve the V-DStP. Instances are 3D grid graphs modeling VLSI chips where we pack relatively many trees, usually 19 or 24, each of which typically contains few terminal nodes (3 or 4). Such grid graphs can be seen as multi-layers graphs where we allow two different kinds of connections. In the *multi-crossed* layers, each node is connected to all its possible neighbors in all directions: the resulting graphs are cubic lattices. The *multi-aligned* layers are similar to the multi-crossed ones but in each layer we allow only connections in one direction, either east-to-west or north-to-south [16, 63]. In table 5.1 we first report some information (type of the layers, size, number of sub-graphs and total number of terminals) concerning each instance and our results. We show the energies achieved by reinforced Max Sum along with the ones of the heuristics "J" and "N" explained in section §4.5. Results are compared to the ones obtained through a state-of-the-art linear programming (LP) technique [16] which is able to provide the energies of the optimal solutions for these instances. In particular we measure the percentage gap between our best energies and the LP ones as

$$Gap\,(our, LP) = \frac{(E_{our} - E_{LP})}{E_{LP}} \cdot 100 \tag{5.5.1}$$

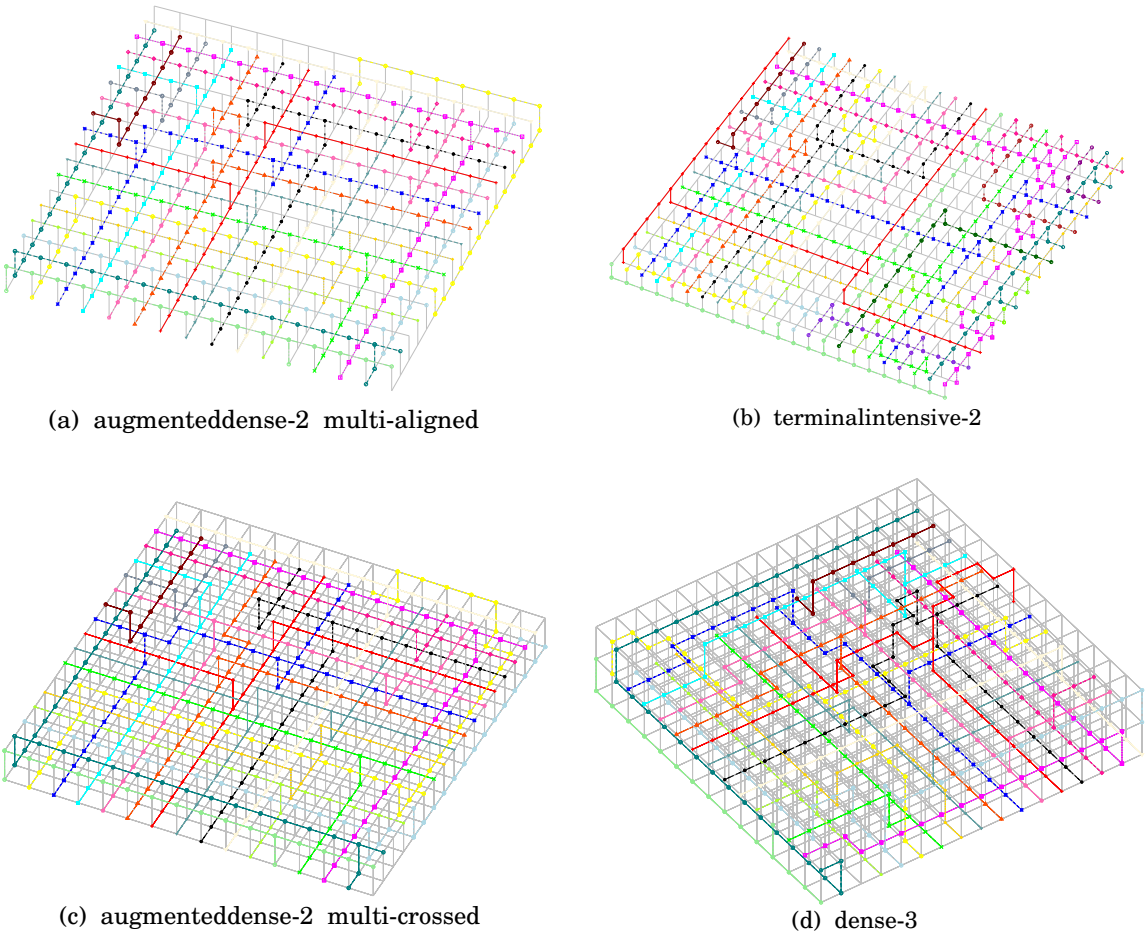
The "-" sign in table 5.1 denotes that no solution has been found. Although we cannot reach better solutions than the ones provided by LP, the gaps are always smaller than 5% and in one case, for the *augmenteddense-2* instance, we output the same solution. We stress that these graphs are very loopy and far from being locally tree-like but nevertheless we achieve good performances thanks to the reinforcement procedure along with the introduction of the modified heuristics. In figure 5.5.1 we display four selected solutions of the set presented in this section.

5.5.2 Fully connected graphs

Here we report the results for the V-DStP on fully connected (FC) graphs of 500 nodes where we aim at packing three trees. Weights of the edges are assigned through two

	Type	**Size**	M	T_{tot}	**"J"**	**"N"**	**MS**	**LP**	**Gap %**
augmenteddense-2	aligned	16x18x2	19	59	504	504	507	504	0 %
augmenteddense-2	crossed	16x18x2	19	59	503	-	-	498	1.0 %
dense-3	crossed	15x17x3	19	59	487	488	485	464	4.0 %
difficult-2	aligned	23x15x2	24	66	535	538	538	526	1.7 %
modifieddense-3	crossed	16x17x3	19	59	492	496	495	479	2.6 %
moredifficult-2	aligned	22x15x2	24	65	542	542	546	522	3.8 %
pedabox-2	aligned	15x16x2	22	56	405	405	405	390	3.8 %
terminalintensive-2	aligned	23x16x2	24	77	599	617	620	596	0.5 %

Table 5.1: Results for circuit layout instances



(a) augmenteddense-2 multi-aligned

(b) terminalintensive-2

(c) augmenteddense-2 multi-crossed

(d) dense-3

Figure 5.5.1: Examples of solutions for the V-DStP on VLSI circuits for multi-aligned, (a) and (b) figures, and multi-crossed, (c) and (d) figures, layouts

different processes that will be made explicit in the following two subsections; terminals are randomly chosen among the nodes of the graphs.

We compare our performances against a "greedy" procedure: first, we sequentially apply the MS algorithm for the single-tree MStP to each communication and then we compute the "greedy" energy as the sum of energies of single-tree solutions. As in the case of the heuristics described in section §5.4 the input graph is carefully modified in order not to use edges in more than one sub-graphs. Notice that this "greedy" procedure is actually hard as the packing problem, since even solving the MStP belongs to NP-hard class of problem; nevertheless this procedure will be useful to underling the benefits carried by the parallel (packing) search against a sequential one. We decide to deal with fully connected graphs because here there always exist configurations of variables such that the hard constrained are satisfied. For instance, to have a candidate solution, it suffices to connect all terminals through a chain.

We report here the values of the energies and gaps $Gap\,(Greedy, MS)$, averaged over about 10^2 instances, as functions of the number of terminal nodes. We run both algorithms, denoted as "MS" and "Greedy", with fixed parameters $D = \{3, 5, 10\}$ and fixed reinforcement factor $\gamma_0 = 10^{-5}$.

5.5.2.1 Independently distributed edge-weights

For these experiments we created several instances of FC graphs where with the edges of the graphs we associate uniformly and independently distributed weights in the interval $[0, 1]$. Results in figure 5.5.2 show that the energies provided by MS algorithm are always smaller then the ones of the greedy solutions. In fact, the energy gaps are always positive for any assignment of terminals and for any depth D; gaps reach their maximum values when the number of terminals, for each communication, lies in the range $[60, 100]$. We notice that energies decrease as we increase the value of the diameter D of the solution; we mention that the computation time, in seconds, required by the simulations (using both strategies) for the largest value of D is orders of magnitude larger then the time needed for $D = 3$.

5.5.2.2 Correlated edge-weights

To underline the benefits carried by the optimized strategy, we run reinforced and greedy reinforced Max-Sum on complete graphs with correlated edge weights. With each node i we assign a uniformly distributed random variable x_i in the interval $[0, 1]$ and for each edge (i, j) we pick a variable $y_{ij} \in (0, 1)$. Then an edge (i, j) will be characterized by a weight $w_{ij} = x_i x_j y_{ij}$. We denote this kind of graphs as "FCW". Results shown in

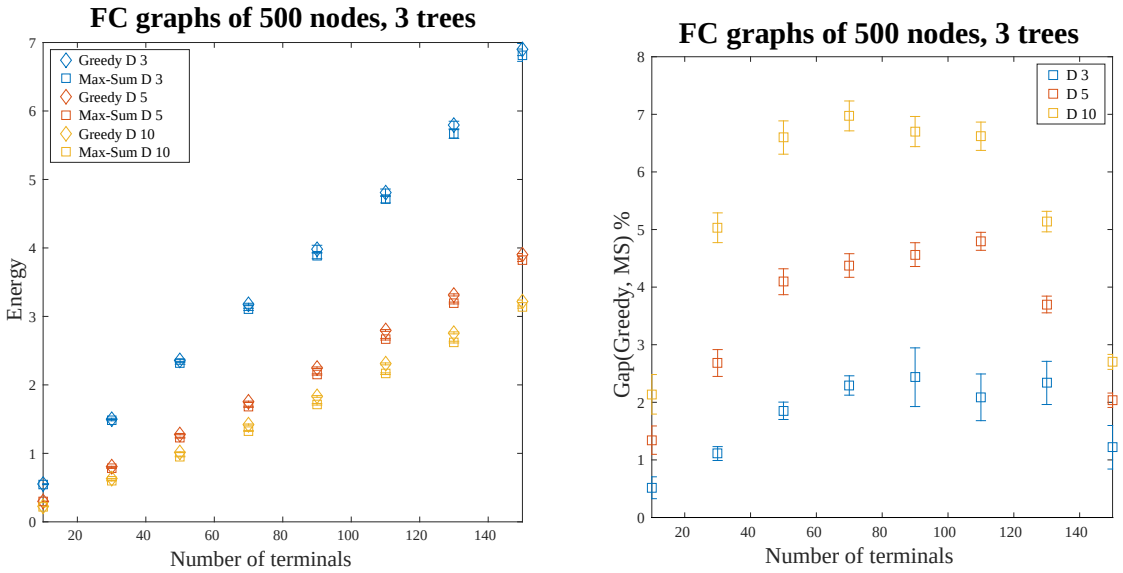


Figure 5.5.2: Average values of the energies and of the energy gaps for fully connected graphs as functions of terminals

figure 5.5.3 reveal that in this particular assignment of the weights, the energy gaps reach very large values (like 60-70 % in some cases) suggesting that the MS algorithm outperform the sequential greedy procedure.

5.6 Discussion

We have shown in this section how to treat two packing variants of the Steiner tree problem, the Vertex-disjoint Steiner tree problem and the Edge-disjoint Steiner tree problem, using an edge-variable formalism similar to the one used in [61]. The methodologies derive from the message passing algorithm at zero-temperature, also known as Max Sum algorithm. We have discussed how to implement three algorithm, one for V-DStP and two for the E-DStP problem; in particular the two concerning the E-DStP scale differently with respect to the system size and can be more or less advantageous depending on the properties of the graph. The edge-based formalism along with the partial information carried by messages before convergence have allowed us to implement fast heuristics able to provide feasible solutions even in hard instances like grid graphs. We have reported several preliminary results of the algorithm for the V-DStP applied to fully connected graphs and to some benchmark VLSI grids on which the optimal solution is known. In the first case MS algorithm always provide better results then the outcomes of a "greedy" procedure reaching energy gaps of 60-70 % for fully connected graphs which

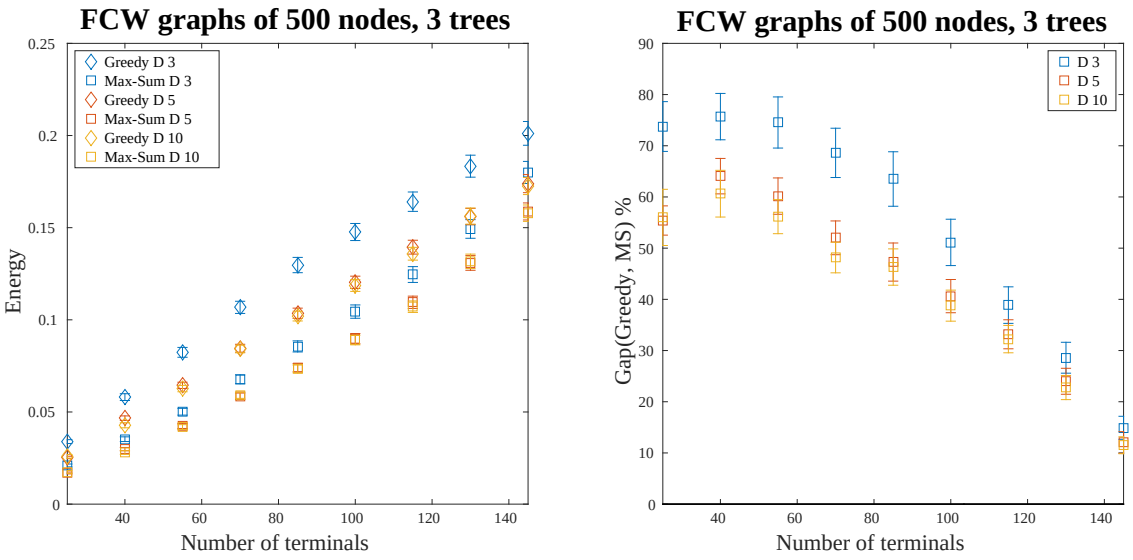


Figure 5.5.3: Average values of the energies and of the energy gaps for fully connected graphs with correlated edge weights as functions of terminals.

edge weights are correlated with one another. In the latter case, energies do not exceed the 5% of the optimal energies computed via linear programming techniques. Further developments regard the efficient implementation of the algorithms for the E-DStP to cope with real-world instances as the VLSI grids.

Chapter 6

EP algorithm for the CS problem

This section is devoted to complete the derivation of the EP algorithm in section 3.5.1 and to report and analyze the preliminary results obtained by EP for the CS problem described in section §2.2.1. In the following we will explicitly compute the moments of the tilted distributions in (3.5.3) for the posterior in (2.2.5). We underline that we are introducing an L_0 regularization differently to the work in [64] where an EP algorithm is presented for a model based on the L_1 regularization. Within this section we will additionally present an approach able to reduce the computing time of EP by a factor N; this procedure will be always applied to all the other inference problems presented in the thesis.

We will show several results for different realizations of the matrix of measurements $\mathbf{F}$ and we will compare our predictions for different values of α and fixed ρ to the ones obtained by BP algorithm. Our purpose is to roughly estimate a critical value for α such that perfect reconstruction is guaranteed or, in other word, we want to investigate the limit of applicability of EP algorithm. Results in this section are very naif as they rely on few simulations; a more rigorous analysis will be contemplated in future works.

6.1 Moments of the tilted distribution for the CS problem

Let us specify the tilted distribution associated with the exact posterior probability in (2.2.5) and let us compute its first two moments. Assuming that each prior $\psi_n(x_n) = (1-\rho)\,\delta(x_n) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{x_n^2}{2\lambda}}$ can be approximated as a Gaussian density $\phi_n(x_n) =$

$\frac{1}{\sqrt{2\pi b_n}} e^{-\frac{(x_n-a_n)^2}{2b_n}}$, the tilted distribution for n^{th} unknown reads

$$Q^{(n)}\left(\boldsymbol{x}|\boldsymbol{y}\right) \propto e^{-\frac{1}{2\Delta}(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})^T(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})} \psi_n\left(x_n\right) \prod_{m\neq n} e^{-\frac{(x_m-a_m)^2}{2b_m}} \tag{6.1.1}$$

where we have not made explicit the normalization constant. As discussed in section 3.5.1 this equation can be further reduced to a multivariate Gaussian times an exact prior $\psi_n\left(x_n\right)$ as

$$Q^{(n)}\left(\boldsymbol{x}|\boldsymbol{y}\right) \propto e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)^T \boldsymbol{\Sigma}_{(n)}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)} \psi_n\left(x_n\right) \tag{6.1.2}$$

where

$$\begin{cases} \boldsymbol{\Sigma}_{(n)}^{-1} & = \frac{1}{\Delta}\mathbf{F}^T\mathbf{F} + \mathbf{D}_{(n)} \\ \boldsymbol{\mu}^{(n)} & = \boldsymbol{\Sigma}_{(n)}\left(\frac{1}{\Delta}\mathbf{F}^T\boldsymbol{y} + \mathbf{D}_{(n)}\boldsymbol{a}\right) \end{cases} \tag{6.1.3}$$

Thus we can compute the two first moments of $Q^{(n)}$as

$$\begin{aligned} \left\langle x_n \right\rangle_{Q^{(n)}} &= \frac{1}{Z_{Q^{(n)}}} \int dx_n x_n e^{-\frac{(x_n-\mu_n)^2}{2\Sigma_{nn}}} \left[(1-\rho)\,\delta\left(x_n\right) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{x_n^2}{2\lambda}}\right] \\ &= \frac{1}{Z_{Q^{(n)}}} \rho \sqrt{\frac{\Sigma_{nn}}{\Sigma_{nn}+\lambda}} \frac{\lambda\mu_n}{\Sigma_{nn}+\lambda} \end{aligned} \tag{6.1.4}$$

$$\begin{aligned} \left\langle x_n^2 \right\rangle_{Q^{(n)}} &= \frac{1}{Z_{Q^{(n)}}} \int dx_n x_n^2 e^{-\frac{(x_n-\mu_n)^2}{2\Sigma_{nn}}} \left[(1-\rho)\,\delta\left(x_n\right) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{x_n^2}{2\lambda}}\right] \\ &= \frac{1}{Z_{Q^{(n)}}} \rho \sqrt{\frac{\Sigma_{nn}}{\Sigma_{nn}+\lambda}} \left[\frac{\lambda\Sigma_{nn}}{\Sigma_{nn}+\lambda} + \left(\frac{\lambda\mu_n}{\Sigma_{nn}+\lambda}\right)^2\right] \end{aligned} \tag{6.1.5}$$

where the normalization constant reads

$$Z_{Q^{(n)}} = (1-\rho)\, e^{\frac{-\lambda\mu_n^2}{2\Sigma_{nn}(\Sigma_{nn}+\lambda)}} + \rho\sqrt{\frac{\Sigma_{nn}}{\Sigma_{nn}+\lambda}} \tag{6.1.6}$$

Now that we have specified the moments of the tilted distributions, we can update the parameters a_n and b_n as (3.5.14). This procedure is iteratively applied to all the unknown variables as to refine the complete set of parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ until we reach the numeric convergence as explained in section 3.5.1.

6.2 Fast computation of EP update equations

The algorithm described in this section requires $\mathcal{O}(N^4)$ iterations which is quite time consuming. In fact, for each one of the N variable, we have to invert the $N \times N$ matrix $\mathbf{\Sigma}_{(n)}^{-1}$ which needs $\mathcal{O}(N^3)$ operations. We show here how to reduce the computing time by a factor N.

Let us consider the following system of equation

$$\begin{cases} \bar{\mathbf{\Sigma}}^{-\mathbf{1}} & = \frac{1}{\Delta}\mathbf{F}^T\mathbf{F} + \mathbf{D} \\ \bar{\boldsymbol{\mu}} & = \bar{\mathbf{\Sigma}}\left(\frac{1}{\Delta}\mathbf{F}^T\boldsymbol{y} + \mathbf{D}\boldsymbol{a}\right) \end{cases} \tag{6.2.1}$$

where $\mathbf{D}$ is a full diagonal matrix of entries $D_{nn} = \frac{1}{b_n}$ or, equivalently, can be expressed as a function of $\mathbf{D}_{(n)}$ as $\mathbf{D} = \mathbf{D}_{(n)} + \frac{1}{b_n}\boldsymbol{e}_n\boldsymbol{e}_n^T$ for $\boldsymbol{e}_n$, the Euclidean versor. The covariance matrix $\bar{\mathbf{\Sigma}}$ and the vector $\bar{\boldsymbol{\mu}}$ parametrize the full approximate distribution $Q(\boldsymbol{x}|\boldsymbol{y})$ in (3.5.4). The elements Σ_{nn} and μ_n relative to the matrix $\mathbf{\Sigma}_{(n)}$and the vector $\boldsymbol{\mu}^{(n)}$ in (6.1.3) can be expressed as functions of the elements $\bar{\Sigma}_{nn}$ and $\bar{\mu}_n$ entering in (6.2.1) as

$$\Sigma_{nn} = \frac{\bar{\Sigma}_{nn}}{1 - \bar{\Sigma}_{nn}\frac{1}{b_n}} \tag{6.2.2}$$

$$\mu_n = \frac{-\frac{1}{b_n}a_n\bar{\Sigma}_{nn} + \bar{\mu}_n}{1 - \bar{\Sigma}_{nn}\frac{1}{b_n}} \tag{6.2.3}$$

Proof Let us left multiply $\mathbf{\Sigma}^{-1}$ to the second equation in (6.2.1) as

$$\left(\frac{1}{\Delta}\mathbf{F}^T\mathbf{F} + \mathbf{D}\right)\bar{\boldsymbol{\mu}} = \mathbf{y} + \mathbf{D}\boldsymbol{a} \tag{6.2.4}$$

for $\mathbf{y} = \frac{1}{\Delta}\mathbf{F}^T\boldsymbol{y}$. The same can be done for the system in (6.1.3) obtaining

$$\left(\frac{1}{\Delta}\mathbf{F}^T\mathbf{F} + \mathbf{D} - \frac{1}{b_n}\boldsymbol{e}_n\boldsymbol{e}_n^T\right)\boldsymbol{\mu}_{(n)} = \mathbf{y} + \left(\mathbf{D} - \frac{1}{b_n}\boldsymbol{e}_n\boldsymbol{e}_n^T\right)\boldsymbol{a} \tag{6.2.5}$$

Performing (6.2.5) - (6.2.4) and left multiplying by $\boldsymbol{e}_n^T$, one finds

$$\mu_n = \frac{-\frac{1}{b_n}a_n\bar{\Sigma}_{nn} + \bar{\mu}_n}{1 - \bar{\Sigma}_{nn}\frac{1}{b_n}} \tag{6.2.6}$$

A similar procedure can be sought to compute Σ_{nn}. Let us define $\boldsymbol{t}$ the solution of equation $\mathbf{\Sigma}_{(n)}^{-1}\boldsymbol{t} = \boldsymbol{e}_n$, that is $\boldsymbol{t}$ is the n^{th} column of $\mathbf{\Sigma}_{(n)}$. Now consider the homogeneous

equation $\bar{\boldsymbol{\Sigma}}^{-1}\bar{\boldsymbol{t}} = \mathbf{0}$ which has solution $\bar{\boldsymbol{t}} = \mathbf{0}$. We write the system of equations:

$$\begin{cases} \left(\frac{1}{\Delta}\mathbf{F}^T\mathbf{F} + \mathbf{D} - \frac{1}{b_n}\boldsymbol{e}_n\boldsymbol{e}_n^T\right)\mathbf{t} = & \mathbf{e}_n \\ \left(\frac{1}{\Delta}\mathbf{F}^T\mathbf{F} + \mathbf{D}\right)\bar{\mathbf{t}} = & \mathbf{0} \end{cases} \tag{6.2.7}$$

If we subtract the first equation to the second and left multiplying by $\boldsymbol{e}_n^T$ we get

$$\Sigma_{nn} = \frac{\bar{\Sigma}_{nn}}{1 - \bar{\Sigma}_{nn}\frac{1}{b_n}} \tag{6.2.8}$$

Notice that this procedure will be apply for all the following EP algorithm described in section §7.2, section §8.2 and section §9.2.

6.3 Preliminary results

In this section we report several results of our EP algorithm for CS problem. Our prediction will be compared to the outcomes of the implementation of the BP algorithm presented in [26] and available in http://aspics.krzakala.org/. We attempt to solve the CS problem for two families of measurement matrices: random matrices having Gaussian independent entries and correlated matrices.

6.3.1 Random matrices

For these experiments we have chosen to infer a signal $\boldsymbol{s}$ of $N = \{50, 100, 200\}$ components having $K = 0.3 \cdot N$ randomly chosen non-zeros; the sparsity of the signal is thus equal to $\rho = 0.3$. We design 5 measurements matrices $\mathbf{F}$ as random matrices of dimension $M \times N$ whose elements are independently distributed random variables picked from a Gaussian distribution of zero mean and unitary variance. We modify M in the range $[0.3 \cdot N, 0.7 \cdot N]$ that is, for a measurement rate $\alpha \in [0.3, 0.7]$. The (inverse) variance of the noise $\frac{1}{\Delta}$ has been taken equal to 10^9 and it is considered as known.

As a first experiment, we attempt to seek a solution to the compressed sensing problem for a single instance of $N = 200$ components. More precisely, for each of the 5 measurement matrices, we create a signal of the pre-defined sparsity and try to solve the inverse problem using both EP and BP algorithms. Results are shown on the top of figure 6.3.1. We can notice that both BP (green line) and EP (cyan line) algorithm fails to retrieve the original signal (red line) for very small values of α but then, for about $\alpha \simeq 0.50$, they achieve a good reconstruction. To better understand the behavior of

both algorithms we perform the same experiment 50 times for each of the 5 matrices and for the three values of N; in the bottom of figure 6.3.1, we report two measures of the quality of these results. In the left plot we show the (normalized) L_2-norm of the difference of the reconstructed signal $\boldsymbol{x}$ (either BP or EP) and the exact one $\boldsymbol{s}$

$$Norm_2\left(\boldsymbol{x}-\boldsymbol{s}\right)=\frac{\left(\boldsymbol{x}-\boldsymbol{s}\right)^T\left(\boldsymbol{x}-\boldsymbol{s}\right)}{\boldsymbol{x}^T\boldsymbol{x}+\boldsymbol{s}^T\boldsymbol{s}} \tag{6.3.1}$$

as a function of α. On the right, we plot the fraction of the "true-positive"

$$TP=\frac{1}{N}\sum_n\left\{\mathbb{I}\left[x_n=0\right]\mathbb{I}\left[s_n=0\right]+\mathbb{I}\left[x_n\neq 0\right]\mathbb{I}\left[s_n\neq 0\right]\right\} \tag{6.3.2}$$

as a function of α. Error bars are computed through the standard error of the measurements. We also report as a vertical green line the critical value of α, called $\alpha_c^{L_1}$ computed according to [30] when the prior is of type L_1. As shown in the plots both BP and EP reach optimal reconstruction for a range of values of α smaller than $\alpha_c^{L_1}$. Concerning the comparison between BP and EP, it seems that EP achieves slightly better results but clearly the number of samples used for these results is insufficient to determined their critical values.

6.3.2 Correlated matrices

In this section we perform the same experiments described in section 6.3.1 with correlated matrices. To ensure that rows of the matrix $\mathbf{F}$ are at the same time correlated and linearly independent, we pick each entry of $\mathbf{F}$ from a multivariate distribution of zero means and a carefully designed $N\times N$ covariance matrix $\mathbf{S}$. The matrix $\mathbf{S}$ has been created as follows. We start from a $k\times N$ matrix $\mathbf{T}$ of independently and Gaussian distributed elements with zero mean and unitary variance, and a diagonal matrix $\mathbf{D}$ having the same kind of (only positive) entries in the diagonal. The covariance matrix is defined as

$$\mathbf{S}=\mathbf{T}^T\mathbf{T}+\mathbf{D} \tag{6.3.3}$$

Notice that the product $\mathbf{T}^T\mathbf{T}$ guarantees that (i) $\mathbf{S}$ is semi-positive definite as its elements correspond to the variances or covariances of the multivariate distribution and (ii) $\mathbf{S}$ is symmetric in agreement with the properties of the covariances. The introduction of $\mathbf{D}$ ensures that $\mathbf{S}$ is a full rank matrix without affecting the signs of the variances.

We present in figure 6.3.2 and in figure 6.3.3 the results for $k\in\{5,20\}$. In both the simulations, BP converges when $N=50$ but for larger values of N does not converge for

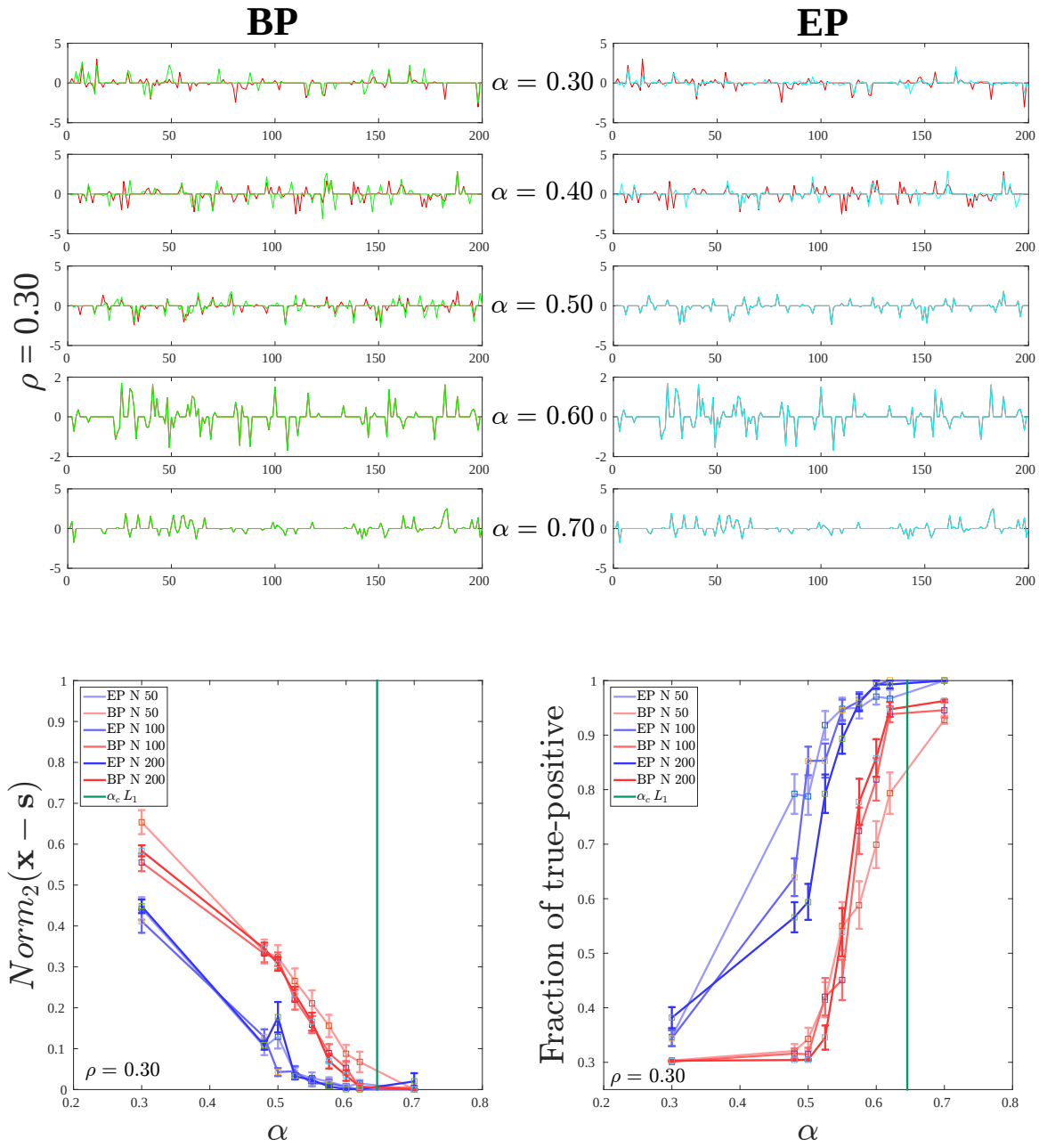


Figure 6.3.1: Single instance (top) and averaged (bottom) results of EP and BP for the CS problem. Random matrix case.

any value of α; in these cases BP predictions are computed using messages in a state of inconsistency. In fact, as shown in the single-instance results on the top of figure 6.3.2 and figure 6.3.3, the recovered signals are quite far from the desired ones. On the bottom of figure 6.3.2 and figure 6.3.3 we can see that both the L_2-norm and the fraction of true-positive remark the inaccuracy of BP results (especially for $N = \{100, 200\}$): the L_2-norm in (6.3.1) is not converging to zero and the fraction of true positive apparently does not reach the value 1 even though α increases up to 0.7. This is quite unexpected; since the absolute values of the non-diagonal elements of $\mathbf{S}$ must behave as $|S_{ij}| \sim k^{1/2}$ for $j \neq i$, we expect that for $N \gg k$, the correlations among the entries of $\mathbf{F}$ should decrease and thus BP should converge.

At difference to BP, EP is able to retrieve the original signal reaching a perfect recovering for $\alpha = 0.60$ as in the case of the random matrices. Correlations within these structured matrices have not affected the performances of EP algorithm.

6.4 Discussion

We have discussed in this chapter how to implement an EP algorithm able to approximate the exact posterior for the CS problem in (2.2.5). We have also shown how to speed up the main loop of the EP algorithm by a factor N with a method that requires only one inversion of an $N \times N$ matrix per iteration. This approach will be applied for all the other EP algorithms presented in this thesis.

We have reported some preliminary results that remark the huge range of applicability of our implementation of the EP approximation for the CS problem. We have seen that the performances of our algorithm seem not to be affected by strong correlations presented in the measurement matrices. This is an important advantage as in real world applications of the CS problem these matrices are often structured and, as we have shown, cavity-based algorithms fail to converge in these cases. In addition, we have shown that the critical value of the measurement rate above which this standard implementation of EP guarantees a good solution is closer to the theoretical limit (the L_0 curve) than the BP one. This allows to compress data in a strong under-sampling regime which can be significantly advantageous in practical applications.

We intend to deeply and rigorously analyze the behavior and the achievable performances of this algorithm in future works.

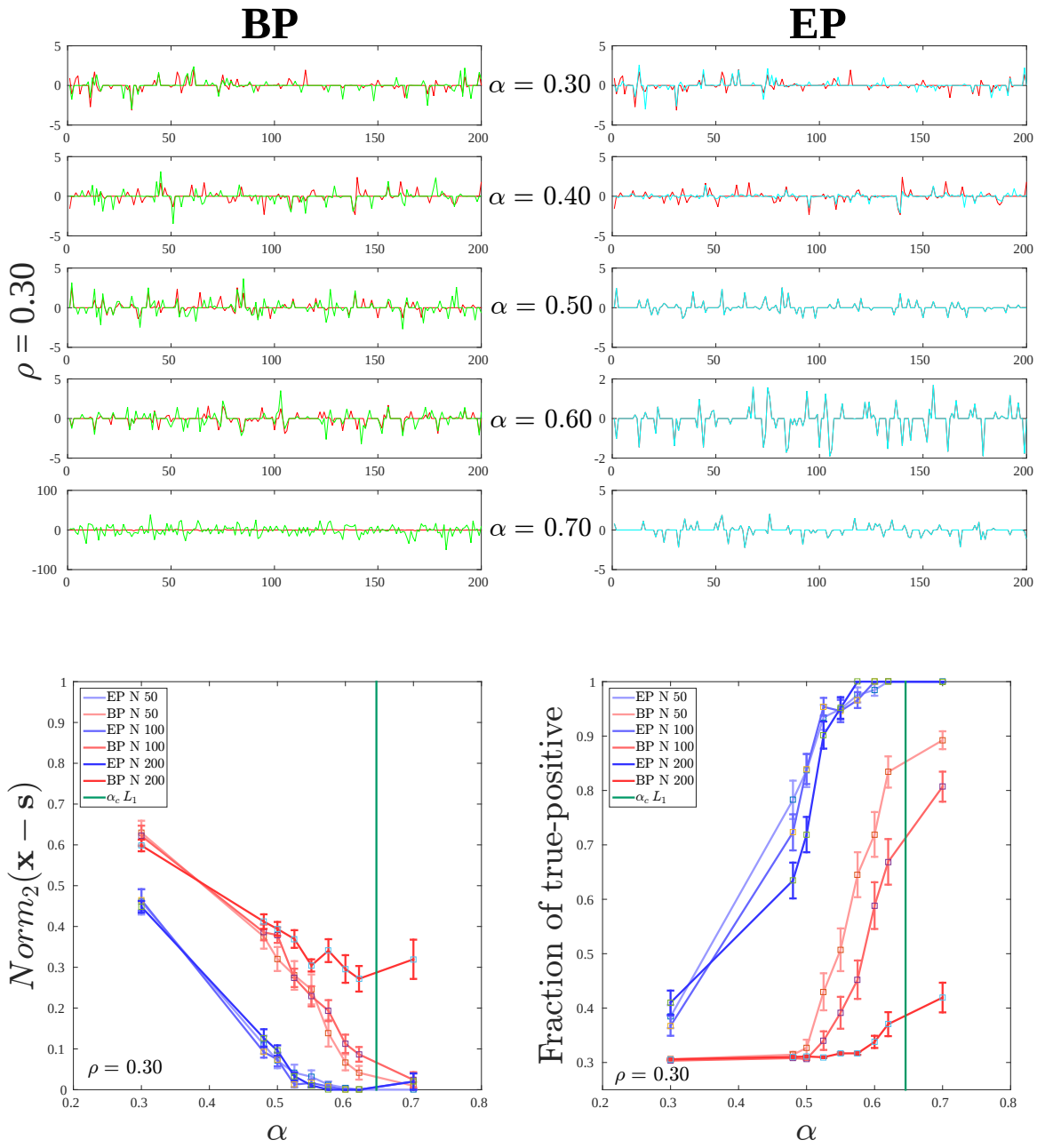


Figure 6.3.2: Single instance (top) and averaged (bottom) results of EP and BP for the CS problem. Structured matrix case with $k = 5$

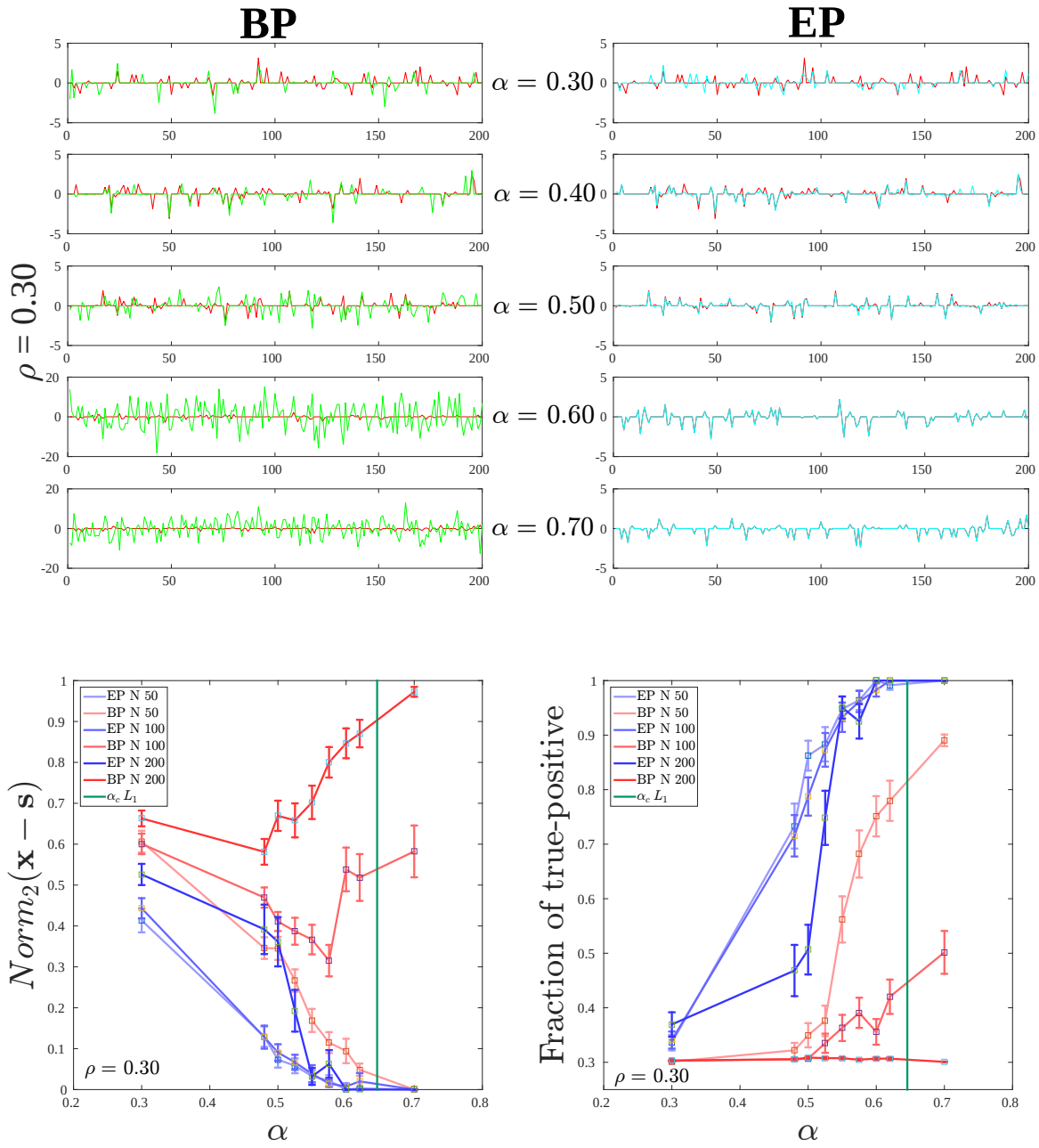


Figure 6.3.3: Single instance (top) and averaged (bottom) results of EP and BP for the CS problem. Structured matrix case with $k = 20$

Chapter 7

EP approximation for inferring metabolic fluxes

As discussed in section §2.2.2 the problem of determining the metabolic fluxes satisfying (2.2.11) and (2.2.12) is hard to solve exactly. Although sampling techniques may provide an estimate of the polytope associated with the solution, it suffers from several drawbacks as illustrated in section §2.2.2. Here we describe an EP based algorithm following the indications in section 3.5.1. The approximation of the marginals provided by this approach turns out to be well comparable to the HR prediction in the asymptotic limit for a set of state-of-the-art large metabolic networks of several kind of organisms. We underline that the EP running time is order of magnitude smaller then the sampling time of HR that, in some cases, is unfeasible.

Moreover, we will explain how to investigate the behavior of fluxes when some marginals are constrained; we show how to modify EP update equations to impose a desired profile to a marginal posterior. We will test this modified version of the algorithm trying to infer the behavior of the metabolic fluxes of Escherichia Coli when its growth rate is constrained. The profile of the biomass has been chosen to fit with real experimental data extracted from [65].

All this section of the thesis is part of the work in [66].

7.1 Bayesian model

In the following we describe how to map the problem in section §2.2.2 into a Bayesian inference problem. Let us define a non-negative energy function underlining the linear constraints in (2.2.11) as

$$H(\boldsymbol{\nu}, \boldsymbol{b}) = \frac{1}{2} (\mathbf{S}\boldsymbol{\nu} - \boldsymbol{b})^T (\mathbf{S}\boldsymbol{\nu} - \boldsymbol{b}) \tag{7.1.1}$$

Notice that the minimum(s) of this energy lies on the configurations of $\boldsymbol{\nu}$ and $\boldsymbol{b}$ satisfying $\mathbf{S}\boldsymbol{\nu} = \boldsymbol{b}$. Equivalently, we can ask what is the probability of observing the intakes/uptakes $\boldsymbol{b}$ given a set of fluxes, or, in other words, what is the likelihood. As we did for the CS problem in section 2.2.1.1 we model it as a Boltzmann distribution

$$P(\boldsymbol{b}|\boldsymbol{\nu}) = \left(\frac{\beta}{2\pi}\right)^{\frac{M}{2}} e^{-\frac{\beta}{2}(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})^T(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})} \tag{7.1.2}$$

where now we set $\beta \to +\infty$ to favor those configurations of fluxes satisfying (2.2.11), i.e. minimize the energy function. Finally we need to specify the prior probability of fluxes: the only prior knowledge that we can include in the model is that they can assume bounded values as indicated in (2.2.12). Thus the prior $P(\boldsymbol{\nu})$ reads

$$P(\boldsymbol{\nu}) = \prod_{n=1}^{N} \frac{\mathbb{I}\left[\nu_n^{min} \leq \nu_n \leq \nu_n^{max}\right]}{\nu_n^{max} - \nu_n^{min}} = \prod_{n=1}^{N} \psi_n(\nu_n) \tag{7.1.3}$$

According to Bayes' theorem in (1.3.1), the posterior probability is given by

$$P(\boldsymbol{\nu}|\boldsymbol{b}) = \frac{1}{P(\boldsymbol{b})} \left(\frac{\beta}{2\pi}\right)^{\frac{M}{2}} e^{-\frac{\beta}{2}(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})^T(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})} \prod_{n=1}^{N} \psi_n(\nu_n) \tag{7.1.4}$$

from which we seek to compute the marginal probability of each flux, that is to calculate $P(\nu_n|\boldsymbol{b})$ for $n = \{1, \ldots, N\}$. Unfortunately, the direct marginalization of (7.1.4) will require the evaluation of high dimensional integrals that, due to the functional form of the priors, is not analytically computable. One way of treating the integration is to approximate each factor of the prior via a Gaussian probability density whose moments are easily evaluable. We present in the next section how to derive an EP algorithm for determining the parameters (the means and the variances) of the Gaussian probability densities of the approximation.

7.2 EP equations

Let us approximate the exact posterior in (7.1.4) with a multivariate Gaussian of the type

$$Q(\boldsymbol{\nu}|\boldsymbol{b}) \propto e^{-\frac{\beta}{2}(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})^T(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})} \prod_{n=1}^{N} \phi_n(\nu_n) \tag{7.2.1}$$

where the functions $\{\phi_n\}_{n=1,\dots,N}$ are Gaussian probability densities of the form $\phi_n(\nu_n) = \frac{1}{\sqrt{2\pi d_n}} e^{-\frac{(\nu_n - a_n)^2}{2d_n}}$ of mean a_n and variance d_n that have to be determined. Our purpose is to estimate these parameters through the EP algorithm discussed in section §3.5.

Let us specify the tilted distribution associated with the approximation in (7.2.1)

$$Q^{(n)}(\boldsymbol{\nu}|\boldsymbol{b}) \propto e^{-\frac{\beta}{2}(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})^T(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})}\psi_n(\nu_n)\prod_{m\neq n}\phi_m(\nu_m) \tag{7.2.2}$$

or, as we have explained in section §6.1 for the CS problem, it can be expressed in standard form as

$$Q^{(n)}(\boldsymbol{\nu}|\boldsymbol{b}) \propto e^{-\frac{1}{2}\left(\boldsymbol{\nu}-\boldsymbol{\mu}^{(n)}\right)^T\boldsymbol{\Sigma}_{(n)}^{-1}\left(\boldsymbol{\nu}-\boldsymbol{\mu}^{(n)}\right)}\psi_n(\nu_n) \tag{7.2.3}$$

where $\boldsymbol{\Sigma}_{(n)}^{-1}$ and $\boldsymbol{\mu}^{(n)}$ satisfy

$$\begin{cases} \boldsymbol{\Sigma}_{(n)}^{-1} &= \beta\mathbf{S}^T\mathbf{S} + \mathbf{D}_{(n)} \\ \boldsymbol{\mu}^{(n)} &= \boldsymbol{\Sigma}_{(n)}\left(\beta\mathbf{S}^T\boldsymbol{b} + \mathbf{D}_{(n)}\boldsymbol{a}\right) \end{cases} \tag{7.2.4}$$

As usual $\mathbf{D}_{(n)}$ is a diagonal matrix whose elements satisfy $D_{mm} = \frac{1}{d_m}$ for $m \neq n$ and $D_{nn} = 0$. To obtain an implementable version of the update equations in (3.5.14) we need to explicit the first and second moments of the distribution in (7.2.3) and thus compute

$$\langle\nu_n\rangle_{Q^{(n)}} \propto \int d\nu_n \nu_n \psi_n(\nu_n)\, e^{-\frac{(\nu_n-\mu_n)^2}{2\Sigma_{nn}}} \tag{7.2.5}$$

$$\left\langle\nu_n^2\right\rangle_{Q^{(n)}} \propto \int d\nu_n \nu_n^2 \psi_n(\nu_n)\, e^{-\frac{(\nu_n-\mu_n)^2}{2\Sigma_{nn}}} \tag{7.2.6}$$

If we notice that the marginal probability $Q^{(n)}(\nu_n|\boldsymbol{b}) \propto \psi_n(\nu_n)\, e^{-\frac{(\nu_n-\mu_n)^2}{2\Sigma_{nn}}}$ is a truncated Gaussian in the interval $[\nu_n^{min}, \nu_n^{max}]$, we can express the two moments in standard form

$$\langle\nu_n\rangle_{Q^{(n)}} = \mu_n + \frac{\mathcal{N}\left(\frac{\nu_n^{min}-\mu_n}{\sqrt{\Sigma_{nn}}}\right) - \mathcal{N}\left(\frac{\nu_n^{max}-\mu_n}{\sqrt{\Sigma_{nn}}}\right)}{\Phi\left(\frac{\nu_n^{max}-\mu_n}{\sqrt{\Sigma_{nn}}}\right) - \Phi\left(\frac{\nu_n^{min}-\mu_n}{\sqrt{\Sigma_{nn}}}\right)}\sqrt{\Sigma_{nn}} \tag{7.2.7}$$

$$\begin{aligned}\langle\nu_n^2\rangle_{Q^{(n)}} - \langle\nu_n\rangle_{Q^{(n)}}^2 = \Sigma_{nn}\Bigg[1 + \frac{\frac{\nu_n^{min}-\mu_n}{\Sigma_{nn}}\mathcal{N}\left(\frac{\nu_n^{min}-\mu_n}{\sqrt{\Sigma_{nn}}}\right) - \frac{\nu_n^{max}-\mu_n}{\sqrt{\Sigma_{nn}}}\mathcal{N}\left(\frac{\nu_n^{max}-\mu_n}{\sqrt{\Sigma_{nn}}}\right)}{\Phi\left(\frac{\nu_n^{max}-\mu_n}{\sqrt{\Sigma_{nn}}}\right) - \Phi\left(\frac{\nu_n^{min}-\mu_n}{\sqrt{\Sigma_{nn}}}\right)} + \\ - \left(\frac{\mathcal{N}\left(\frac{\nu_n^{min}-\mu_n}{\sqrt{\Sigma_{nn}}}\right) - \mathcal{N}\left(\frac{\nu_n^{max}-\mu_n}{\sqrt{\Sigma_{nn}}}\right)}{\Phi\left(\frac{\nu_n^{max}-\mu_n}{\sqrt{\Sigma_{nn}}}\right) - \Phi\left(\frac{\nu_n^{min}-\mu_n}{\sqrt{\Sigma_{nn}}}\right)}\right)^2\Bigg]\end{aligned} \tag{7.2.8}$$

where $\mathcal{N}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ is the standard normal distribution of mean 0 and unitary variance and $\Phi(x) = \int_{-\infty}^{x} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$ is its cumulative. Unfortunately, the above exact expressions suffer from numeric instabilities when the argument $|x| \to +\infty$. Here the difference $\Phi\left(\frac{\nu_n^{max} - \mu_n}{\sqrt{\Sigma_{nn}}}\right) - \Phi\left(\frac{\nu_n^{min} - \mu_n}{\sqrt{\Sigma_{nn}}}\right)$ is not numerically appreciable and the ratios in (7.2.7) (7.2.8) are undetermined. To overcome this problem it is useful to replace the cumulative $\Phi(x)$ in (7.2.7)(7.2.8) with its expansion in the limit $|x| \to +\infty$ that is, up to the 5^{th} order,

$$\Phi(x) \simeq \frac{1}{2} - \mathcal{N}(x)\left[\frac{1}{x} - \frac{1}{x^3} + \frac{3}{x^5} - o\left(\frac{1}{x^7}\right)\right] \tag{7.2.9}$$

The threshold value of $|x|$ over which the exact expressions are replaced by the ones obtained from the expansion, has been chosen *ad-hoc* after several numerical trials.

7.3 Update equations for a constrained marginal

Consider the problem outlined in section 2.2.2.1 and let us investigate how to deal with this case in the Bayesian framework introduced in section §7.1. Let us define $f(\nu_i)$ as the (marginal) posterior of the experimental measured flux i. How the posteriors of other fluxes would modify in order to fit with the experimental results? According to maximum entropy principle [11] the most unconstrained distribution which is consistent with the experiment, the prior distribution in (7.1.3) and flux conservation in (2.2.11), is

$$P(\boldsymbol{\nu}|\boldsymbol{b}) \propto e^{-\frac{\beta}{2}(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})^T(\mathbf{S}\boldsymbol{\nu}-\boldsymbol{b})} \prod_{n=1}^{N} \mathbb{I}\left[\nu_n^{inf} \leq \nu_n \leq \nu_n^{sup}\right] g(\nu_i) \tag{7.3.1}$$

for $\beta \to \infty$ and $g(\nu_i)$ the (exponential of the) function of unknown Lagrange multipliers. This function has to be determined in order for the constraint

$$f(\nu_i) = \int \prod_{n \neq i} d\nu_n P(\boldsymbol{\nu}|\boldsymbol{b}) \tag{7.3.2}$$

to be satisfied. Let us consider the case in which $f(\nu_i)$ can be reasonably fitted by a Gaussian distribution $\mathcal{N}(\nu_i|a_i^{exp}, d_i^{exp})$, then it is sufficient to consider also a Gaussian $g(\nu_i) = \mathcal{N}(\nu_i|a_i, d_i) = \phi_i(\nu_i; a_i, d_i)$ parametrized by a_i and d_i. We will show here how to modify the EP algorithm for flux i in order to determine the two parameters; this part can be included in the main loop of original EP without affecting the computing time. Assuming as before that the prior of each flux $n \neq i$ can be approximated as a Gaussian

density $\phi_n(\nu_n; a_n, d_n)$ of (unknown) parameters a_n and d_n, also to be determined, we must impose that

$$\mathcal{N}(\nu_i|a_i^{exp}, d_i^{exp}) \propto \mathcal{N}(\nu_i|a_i, d_i) \int \prod_{n\neq i} d\nu_n Q(\boldsymbol{\nu}|\boldsymbol{b}) \tag{7.3.3}$$

$$\propto \phi_i(\nu_i; a_i, d_i)\, e^{-\frac{(\nu_i-\mu_i)^2}{2\Sigma_{ii}}} \tag{7.3.4}$$

where the distribution $Q(\boldsymbol{\nu}|\boldsymbol{b})$ is the one in (7.2.1). Since both the left-hand side and the right-hand side of (7.3.4) contain Gaussian distributions, the relations for a_i and d_i can be easily computed and take the form

$$\begin{cases} d_i &= \left(\frac{1}{d_i^{exp}} - \frac{1}{\Sigma_{ii}}\right)^{-1} \\ a_i &= d_i\left(\frac{a_i^{exp}}{d_i^{exp}} - \frac{\bar{\nu}_i}{\Sigma_{ii}}\right) \end{cases} \tag{7.3.5}$$

This expression is exactly the same in (3.5.14) if we replace the mean and the variance of the tilted distribution with the experimental ones.

7.4 Results

In this section we report a representative part of the results reported in [66].

7.4.1 Two large scale metabolic networks

Let us compare our estimate of the marginal probabilities evaluated according to (7.2.3) against the HR distributions obtained through the sampling procedure explained in section 3.6.2.1. We apply both EP and HR algorithms to two models chosen from the Bigg Model database [67]: *GLUnorm* model [68] describing the *Glutamatergic neuron* and *RECON1* model [69] for *Homo Sapiens*. In particular the last one is very complex as it involves 2469 reactions among 1587 metabolites.

Before applying the two methods, we pre-process the stoichiometric matrix in order to erase all the fluxes that can be only produced or only degraded. Then, we run EP and HR (more precisely an optimized implementation of HR, called *optGpSampler* [60]) both implemented in Matlab codes.

Regarding the HR simulations we set the number of sampled points to be equal to 10^4 for an increasing number of explored configurations T from 10^4 to 10^7 for the *GLUnorm* and up to $T \sim 10^9$ points for *RECON1*. Differently, numerical convergence of EP depends on the refinement of parameters $\boldsymbol{a}$ and $\boldsymbol{d}$ or, more precisely, on the estimate

of the marginal distributions of fluxes as discussed in (3.5.15). The parameter β has been set to 10^{10} for *GLUnorm* and 10^9 for *RECON1* model.

To quantitatively compare the two approaches we report here the scatter plots of variances and means of the approximate marginals computed via HR and EP along with a measure of the correlation among the two sets of parameters. This quantity is computed through the Pearson product-moment correlation coefficient defined as

$$\rho_{X_{HR}X_{EP}} = \frac{cov\left(X_{HR}, X_{EP}\right)}{\sigma_{X_{HR}}\sigma_{X_{EP}}} \tag{7.4.1}$$

where X_{HR} and X_{EP} can be the population of means or variances computed according to HR prediction and our algorithm respectively; *cov* and σ denote the covariance and the standard deviation of the sets under investigation.

In figures 7.4.1 and 7.4.2 we show the results for *Glutamatergic neuron* and *Homo Sapiens* respectively. Each figure shows several plots:

- **Left plots.** On the bottom-left we plot the running time of both algorithm and on the top-left the correlation coefficients for the means and the variances of the marginal posteriors. The two measures are plotted as functions of the configurations T explored by HR. As shown in these plots, we can notice that to reach correlation close to 1 we need to fairly sample the polytope by employing a computing time that is always several orders of magnitude larger than the EP running time. The most significant time-gap appears in figure 7.4.2 where EP takes hours to converge at difference to HR which needs several days to finish the sampling.

- **Scatter plots.** The rest of the figures show the scatter plot of the means (top) and the variances (bottom) in different regimes. The $y-$axis hosts the means (variances) of EP predictions plotted against the HR means (variances) for increasing T as specified in the $x-$axis. It is clear that as HR explores more configurations as the values of the means and variances approach the EP estimates. This is particularly clear for the *Glutamatergic neuron* in figure 7.4.1.

7.4.2 Escherichia Coli metabolism for constrained growth rate

Consider the problem presented in section 2.2.2.1 and let us test the (modified) EP algorithm shown in section §7.3 on the *iJR904* model [70] of *Escherichia Coli.* As a matter of example we decide to fix the biomass of Escherichia Coli as indicated in the growth rate labeled as *Glc (P5-ori)* reported in figure 3(a) of [65]. The profile can be

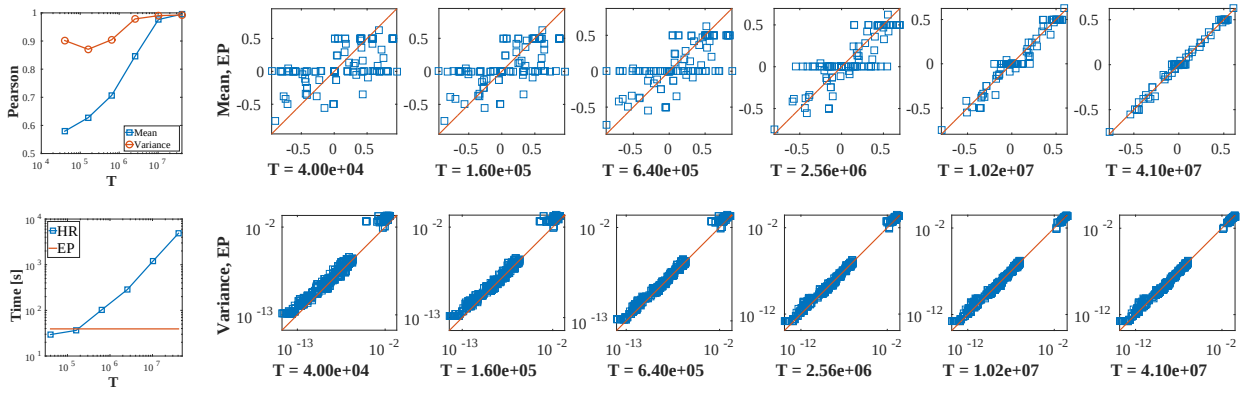


Figure 7.4.1: Comparison between EP and HR results for *Glutamatergic neuron*

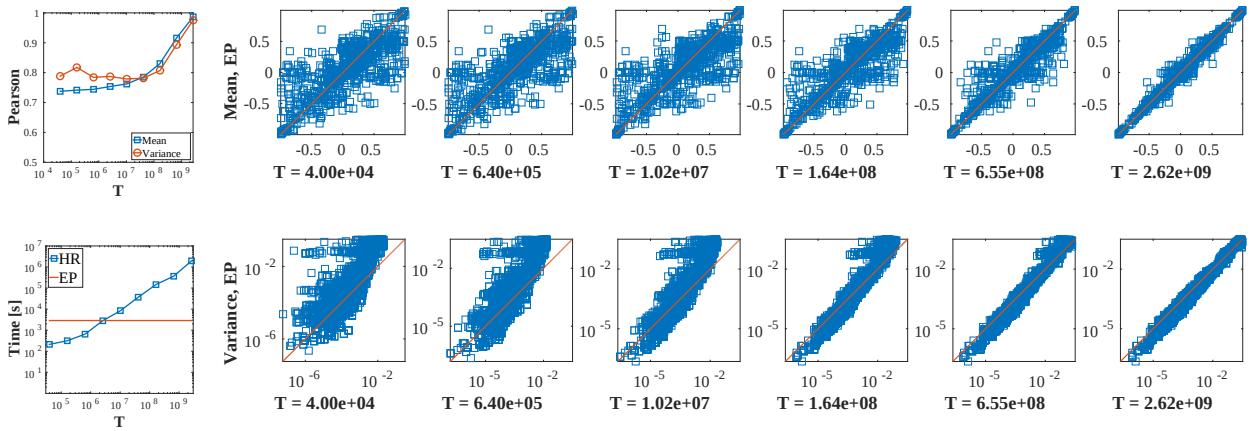


Figure 7.4.2: Comparison between EP and HR results for *Homo Sapiens*

***EX_glc(e)* lower bound**	**Maximum value of the biomass**
-343	11.00
-243	8.00
-143	5.01
-43	2.02

Table 7.1: Maximum value of the biomass (computed through FBA) when we fix the lower bound of *EX_glc(e)* flux

well fitted by a Gaussian probability density of mean $0.92\,\mathrm{h}^{-1}$ and variance $0.0324\,\mathrm{h}^{-2}$. We have chosen the *iJR904* model because, in principle, it can catch the main features of the so-called *minimal substrate* culture medium where these bacteria lived. The bounds on the glucose exchange flux *EX_glc(e)* has been designed in a way that the maximum allowed biomass flux (computed in terms and units of growth rate by FBA, about $2\,\mathrm{h}^{-1}$) contained all experimental values of the used profile. As shown in table 7.1, the lower bound of the exchanged glucose that guarantees the desired threshold shift of the biomass is $-43\,\mathrm{mmol}\ (\mathrm{gDW})^{-1}\,\mathrm{h}^{-1}$.

Then we apply EP algorithm to the modified *iJR904* model when (i) we do not impose any additional constraint and (ii) we fix the experimental profile of the growth rate. As desired, in the second case, the growth rate marginal is exactly the experimental one, but other fluxes changed their marginal probabilities. Let us identify those fluxes that have been more affected by the constrained growth rate. We report in figure 7.4.3 the plot of the ratio between the means (figure 7.4.3 (a)) and the variances (figure 7.4.3 (b)) in the unconstrained case and in the constrained case. In figure 7.4.3 (a) the $x-$axis is the logarithm of the absolute value of the unconstrained means to differentiate those fluxes having means close to zero and all the other cases. The ratios of the variances are instead plotted as a function of the unconstrained variances in semi-log scale. We have reported the name of the reactions with the most significant changes; for instance, the marginal of the *TKT2* reaction has reduced its mean of more than one third, while many reactions involving *aspartame* have significantly lowered their variances.

7.4.3 Red blood cell

In figure 7.4.4 we report the 46 marginal posterior distributions of all the fluxes of the red blood cell model in [71]. The blue bars are obtained through HR sampling while the orange line and the red line are the profiles of the marginals according to BP algorithm implemented in [72] and to EP respectively. We notice that the EP estimates are always

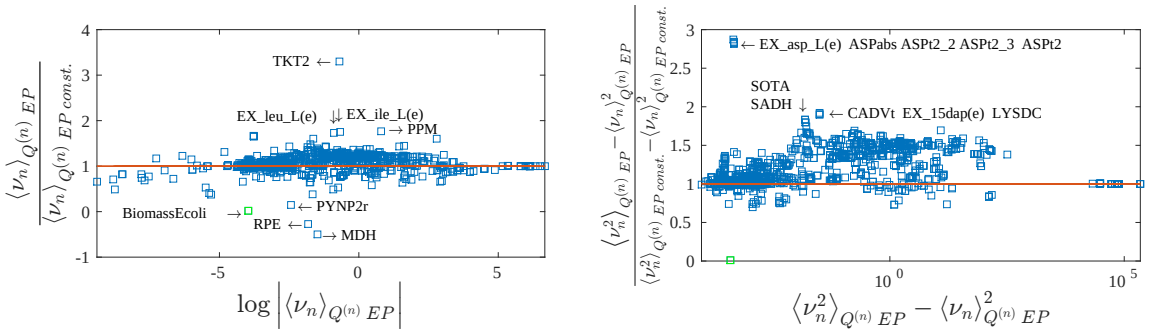


Figure 7.4.3: Comparison between the means (a) and variances (b) of the marginal probability densities for all the fluxes computed without the additional constraint (unconstrained case) and with the constrained on the biomass (constrained case). The green point indicates the biomass flux.

very similar to the marginals predicted by HR and that, for several fluxes (for instance *AK*, *AMPase* and *NADP*), the profiles are much more accurate than the BP ones.

7.5 Discussion

In this chapter we have discussed in detail how to express the problem of describing the space of solution of metabolic fluxes in a Bayesian framework and how to approximate the marginal posterior of each flux as a truncated Gaussian distribution. The parameters of the approximation have been derived through the EP algorithm. The comparison of the first and second moments of the marginal distribution estimated through EP and HR shows that in almost all cases (up to the number of configurations explored by HR in the available time, sometimes up to 20 days), the difference between HR and EP results is monotonically decreasing with the number of configurations. This suggests that EP results are to be preferred over HR results for any smaller number of samples than the ones we analyzed. Thus, if the assumption of the exactness of HR in the asymptotic limit is correct, our results resemble the "true" marginals; we underline that we do not know the "ground-truth" related to this problem.

We have shown how to investigate the behavior of metabolic fluxes in the cases of a constrained flux. This is quite remarkable as it appears that sampling techniques are not able to cope to this different problem in an efficient way. We have also reported a complete list of marginals relative to the red blood cells fluxes in comparison with a state-of-the-art implementation of the BP algorithm and HR sampling. Our results are clearly more similar to the HR estimates then the BP ones.

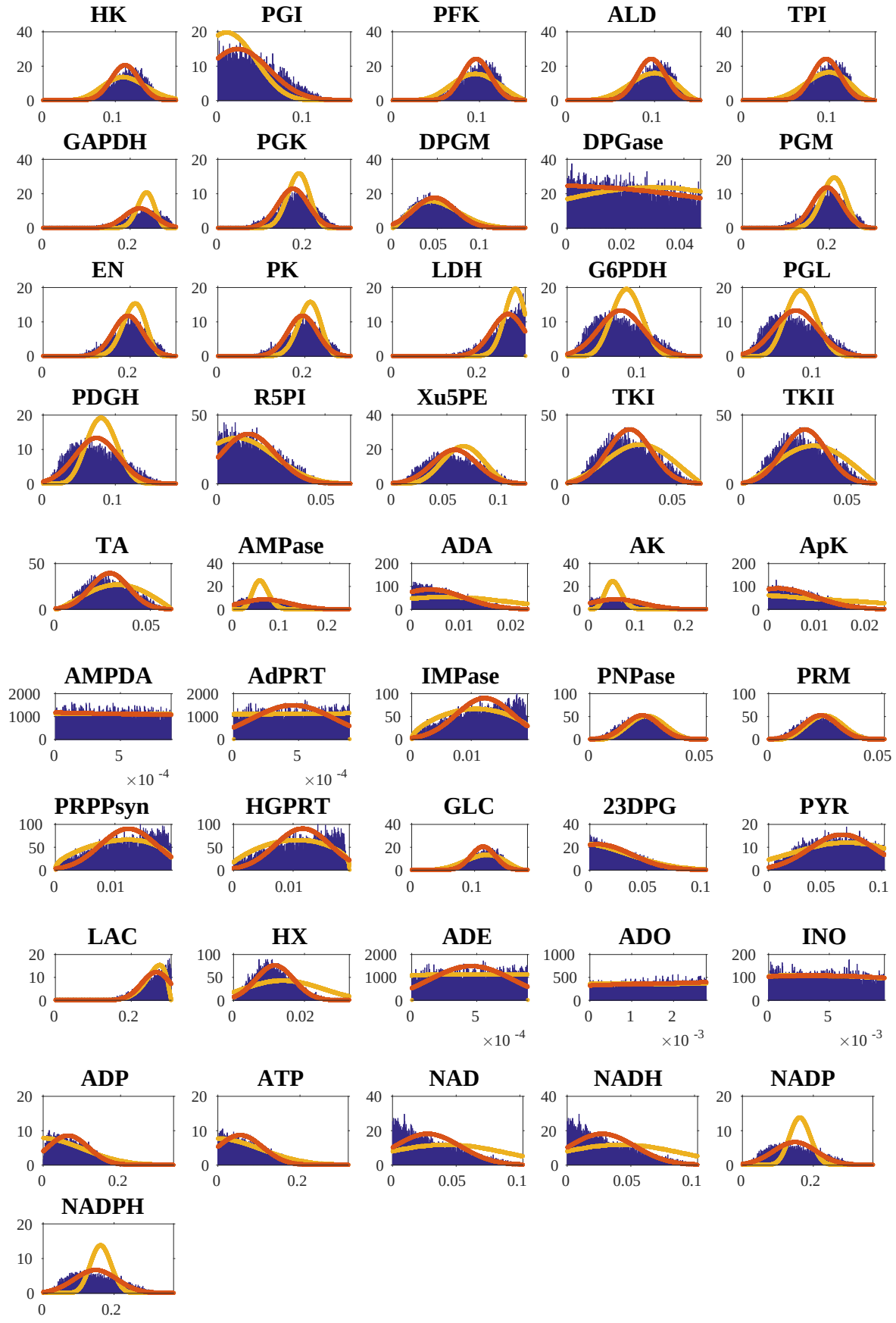


Figure 7.4.4: Marginal probability densities of the fluxes of red blood cell estimate by HR (blue bars), BP (orange line) and EP (red line)

In real cases variances of the marginal distributions can span several orders of magnitude. As a consequence also the variances of the approximation need to allow both very small and huge values. To prevent the numeric problems that may arise, we allow parameters $\boldsymbol{d}$ to vary in a finite range of values, in our case $[10^{-50}, 10^{50}]$, with, in principle, the drawback of limiting the set of allowed Gaussian densities of the approximation. Fortunately, we have observed that at convergence the value of $\boldsymbol{d}$ vary in the range $[10^{-10}, 10^{48}]$ so the interval is sufficient to catch all the flux profiles presented in this work.

The parameter β can be seen as the inverse-variance of a Gaussian noise affecting the stoichiometric equations. The nature of this noise could depend on the spatial heterogeneity of the cell or on real thermal noise. In this case, an optimization of the free energy with respect to β can in principle lead to better predictions.

We underline that the relevance of EP algorithm is twofold. First, its predictive accuracy in a fraction of the computing time with respect to state-of-the-art alternatives allows for the study of much larger models, such as the ones describing the joint metabolism of symbiotic systems of different organisms. Second, the analytic nature of the approach makes it specially adapt to accommodate additional constraints in an efficient way. Besides allowing to incorporate evidence from experiments (such as the distribution of cell growth rate), this point is of utmost importance for the study of new and more detailed models models of cell metabolism that e.g. may add additional constraints to encode information from regulatory processes. The work presented in this chapter opens new research perspectives in the study of metabolic fluxes that will be contemplated in future developments.

Chapter 8

Inferring active regions in EEG inverse problem

As mentioned in section §2.2.3 the inverse problem of the EEG can be described by a model whose formalism is equivalent to the CS problem. However, current densities must satisfy additional constraints that enforce a proper spatial localization of the sources. To group together those currents that belong to the same functional area we will use the so-called dictionary which enables the partition of the electric lead field according to some atlas of the brain. We will see how to exploit the EP algorithm for the CS problem explained in section §6.1 to predict active regions within the brain in this modified scenario.

8.1 A Bayesian model for EEG inverse problem

Let us model the problem introduced in section §2.2.3 in a Bayesian framework. Following the same argument of section 2.2.1.1, we assume that the noise $\boldsymbol{\varepsilon}$ affecting the voltage measurements $\boldsymbol{v}$ is a collection of independently distributed Gaussian random variables of zero mean and variance $\frac{1}{\beta}$. The likelihood of measuring $\boldsymbol{v}$ given a set of currents $\boldsymbol{j}$ is

$$P\left(\boldsymbol{v}|\boldsymbol{j}\right) = \left(\frac{\beta}{2\pi}\right)^{N_e} e^{-\frac{\beta}{2}(\boldsymbol{v}-\mathbf{K}\boldsymbol{j})^T(\boldsymbol{v}-\mathbf{K}\boldsymbol{j})} \tag{8.1.1}$$

Instead of encoding the spatial localization of the currents in the design of the prior, we can add the structural information of the N_a functional areas at the level of the matrix $\mathbf{K}$. Let us define an auxiliary matrix, the dictionary $\mathbf{A}$, of dimension $N_g \times N$ where $N \in \left[N_a, 2^{N_a}\right]$. To give an example of how this matrix works, let us suppose

$N = N_a$. Each column i of the matrix $\mathbf{A}$ has non-zero entries in correspondence of those j-components which belong to area i. In particular the values of the non-zero entries can shape a Gaussian profile to ensure a smooth decay forward non-active regions. If we now define $\tilde{\mathbf{K}}_{N_e \times N_a} = \mathbf{K} \cdot \mathbf{A}$ we can seek a vector $\tilde{\boldsymbol{j}}$ such that $\boldsymbol{v} - \tilde{\mathbf{K}} \cdot \tilde{\boldsymbol{j}} = \boldsymbol{\varepsilon}$; of course this constraint is equivalent to the one enforced in (8.1.1). Assuming that only few areas provide to the electrical activity of the brain for fixed time, the signal $\tilde{\boldsymbol{j}}$ can be considered as K sparse, with a fraction $\rho = \frac{K}{N_a}$ of non-zero entries. Thus we can write the prior for vectors $\tilde{\boldsymbol{j}}$ as

$$P\left(\tilde{\boldsymbol{j}}\right) = \prod_n \left[(1-\rho)\, \delta\left(\tilde{j}_n\right) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{1}{2\lambda}\tilde{j}_n^2} \right] \tag{8.1.2}$$

The posterior now reads

$$P\left(\tilde{\boldsymbol{j}} | \boldsymbol{v}, \mathbf{A}, \rho, \lambda, \beta\right) \propto \left(\frac{\beta}{2\pi}\right)^{N_e} e^{-\frac{\beta}{2}\left(\boldsymbol{v}-\tilde{\mathbf{K}}\cdot\tilde{\boldsymbol{j}}\right)^T\left(\boldsymbol{v}-\tilde{\mathbf{K}}\cdot\tilde{\boldsymbol{j}}\right)} \prod_n \left[(1-\rho)\, \delta\left(\tilde{j}_n\right) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{1}{2\lambda}\tilde{j}_n^2} \right] \tag{8.1.3}$$

where we have underlined the parameters dependency of the model we are using. Thus we can solve the CS problem for the auxiliary variables $\tilde{\boldsymbol{j}}$, or equivalently, in the "space of the areas". Notice that adding to the columns of $\mathbf{A}$ combinations of such areas, we can recover solutions in which also combinations of different regions can be responsible to voltages on the scalp. Smoothness in the "real space" can also be introduced in the values of the entries of the dictionary along with the parameter λ of the prior. Once the CS problem has been solved, we can easily recover the original distributions through the relation $\boldsymbol{j} = \mathbf{A} \cdot \tilde{\boldsymbol{j}}$.

As for the CS problem, the computation of marginal distributions from (8.1.3) is intractable. Notice that in principle one can use the BP algorithm for continuous variables in section 3.4.1; however, the rows of a (realistic) matrix $\mathbf{K}$ are strongly correlated and BP fails to converge.

8.2 EP algorithm

We show here how to approximate the distribution in (8.1.3) applying the EP algorithm in section 3.5.1 (in the following we will simplify the notation using j_n instead of $\tilde{j}_n$). As standard we need to compute the first two moments of the tilted distribution associated with the posterior in (8.1.3), that is

$$Q^{(n)}\left(\boldsymbol{j}|\boldsymbol{v}\right) \propto e^{-\frac{\beta}{2}(\boldsymbol{v}-\mathbf{K}\boldsymbol{j})^T(\boldsymbol{v}-\mathbf{K}\boldsymbol{j})} \psi_n\left(j_n\right) \prod_{m\neq n} e^{-\frac{(j_m-a_m)^2}{2b_m}} \qquad n \in \{1,\ldots,N\} \tag{8.2.1}$$

where we have used a short notation for the exact prior of the n^{th} current

$$\psi_n\left(j_n\right) = \left[\left(1-\rho\right)\delta\left(j_n\right) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{1}{2\lambda}j_n^2}\right] \tag{8.2.2}$$

Notice that the functional form of (8.2.1) is equivalent to (6.1.1) and therefore the mean and variance are equal to

$$\begin{aligned}\left\langle j_n\right\rangle_{Q^{(n)}} &= \frac{1}{Z_{Q^{(n)}}}\int dj_n j_n e^{-\frac{(j_n-\mu_n)^2}{2\Sigma_{nn}}}\left[\left(1-\rho\right)\delta\left(j_n\right) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{j_n^2}{2\lambda}}\right] &(8.2.3)\\ &= \frac{1}{Z_{Q^{(n)}}}\rho\sqrt{\frac{\Sigma_{nn}}{\Sigma_{nn}+\lambda}}\frac{\lambda\mu_n}{\Sigma_{nn}+\lambda} &(8.2.4)\end{aligned}$$

$$\begin{aligned}\left\langle j_n^2\right\rangle_{Q^{(n)}} &= \frac{1}{Z_{Q^{(n)}}}\int dj_n j_n^2 e^{-\frac{(j_n-\mu_n)^2}{2\Sigma_{nn}}}\left[\left(1-\rho\right)\delta\left(j_n\right) + \frac{\rho}{\sqrt{2\pi\lambda}} e^{-\frac{j_n^2}{2\lambda}}\right] &(8.2.5)\\ &= \frac{1}{Z_{Q^{(n)}}}\rho\sqrt{\frac{\Sigma_{nn}}{\Sigma_{nn}+\lambda}}\left[\frac{\lambda\Sigma_{nn}}{\Sigma_{nn}+\lambda} + \left(\frac{\lambda\mu_n}{\Sigma_{nn}+\lambda}\right)^2\right] &(8.2.6)\end{aligned}$$

where the normalization constant reads

$$Z_{Q^{(n)}} = \left(1-\rho\right) e^{\frac{-\lambda\mu_n^2}{2\Sigma_{nn}(\Sigma_{nn}+\lambda)}} + \rho\sqrt{\frac{\Sigma_{nn}}{\Sigma_{nn}+\lambda}} \tag{8.2.7}$$

The terms μ_n and Σ_{nn} now satisfy the relation

$$\begin{cases}\boldsymbol{\Sigma}_{(n)}^{-1} &= \beta\mathbf{K}^T\mathbf{K} + \mathbf{D}_{(n)}\\ \boldsymbol{\mu}^{(n)} &= \boldsymbol{\Sigma}_{(n)}\left(\beta\mathbf{K}^T\boldsymbol{v} + \mathbf{D}_{(n)}\boldsymbol{a}\right)\end{cases} \tag{8.2.8}$$

8.3 Inference of the parameters

The model that we have shown in section §8.1 depend on three parameters, β, ρ and λ, along with the dictionary matrix. These parameters have to be tuned in order to reproduce the best possible estimate of the currents. We stress that, in this particular problem, β and ρ are actually related to some physical quantities: the first one estimates the variance of the noise affecting the measurements while the second one (for a dictionary having a number of columns equal to the number of areas) is a measure of "sparsity" in the sense that it provides the fraction of active regions within the brain.

In statistical inference, there exist several strategies to infer the parameters of a model along with the inference of a certain set of variables; we mention, for instance, the Expectation Maximization (EM) [73] strategy for VB methods and Gradient Descent (GD) methods [74]. We have shown in section 3.5.2 that EP fixed point corresponds to the stationary point of the free energy functional in (3.5.17) which approximates the exact free energy. To estimate the three parameters introduced above, we will show how to design a GD scheme over the (approximate) free energy $F_{EP}(\rho, \lambda, \beta)$.

Let us first explicitly compute the free energy for this problem:

$$F_{EP} = (N_g - 1) \log Z_Q - \sum_n \log \int d^N \boldsymbol{j} Q^{(n)}(\boldsymbol{j}|\boldsymbol{v}) \tag{8.3.1}$$

where $Q^{(n)}(\boldsymbol{j}|\boldsymbol{v})$ is the tilted distribution in (8.2.1) and

$$Z_Q = \int d^N \boldsymbol{j} e^{-\frac{\beta}{2}(\boldsymbol{v}-\mathbf{K}\boldsymbol{j})^T(\boldsymbol{v}-\mathbf{K}\boldsymbol{j})} \prod_n e^{-\frac{(j_n - a_n)^2}{2b_n}} \tag{8.3.2}$$

Performing the two integrals one finds

$$\begin{aligned} F_{EP} =& \frac{N_e}{2}(\log 2\pi - \log \beta) + \frac{(N_g - 1)}{2}\left(\log\det \bar{\boldsymbol{\Sigma}} + \bar{\boldsymbol{\mu}}^T \bar{\boldsymbol{\Sigma}}^{-1} \bar{\boldsymbol{\mu}}\right) + \frac{1}{2}\beta \boldsymbol{v}^T \boldsymbol{v} \\ & - \frac{1}{2}\sum_n \left\{ \left(\log\det \boldsymbol{\Sigma}_{(n)} - \log \Sigma_{nn} + \boldsymbol{\mu}^{(n)T} \boldsymbol{\Sigma}_{(n)}^{-1} \boldsymbol{\mu}^{(n)}\right) + \right. \\ & \left. - \log \left[e^{-\frac{\mu_n^2}{2\Sigma_{nn}}}(1-\rho) + e^{-\frac{\mu_n^2}{2(\lambda + \Sigma_{mm})}} \rho \sqrt{\frac{\Sigma_{nn}}{\lambda + \Sigma_{nn}}} \right] \right\} \end{aligned} \tag{8.3.3}$$

where the matrix $\boldsymbol{\Sigma}_{(n)}$ and the vector $\boldsymbol{\mu}^{(n)}$ are defined for each n^{th} tilted distribution and satisfy (8.2.8). The matrix $\bar{\boldsymbol{\Sigma}}$ and the vector $\bar{\boldsymbol{\mu}}$ are instead defined for the full approximation as

$$\begin{cases} \bar{\boldsymbol{\Sigma}}^{-1} & = \beta \mathbf{K}^T \mathbf{K} + \bar{\mathbf{D}} \\ \bar{\boldsymbol{\mu}} & = \bar{\boldsymbol{\Sigma}}\left(\beta \mathbf{K}^T \boldsymbol{v} + \bar{\mathbf{D}} \boldsymbol{a}\right) \end{cases} \tag{8.3.4}$$

in which the matrix $\bar{\mathbf{D}}$ is full diagonal of elements $\bar{D}_{nn} = \frac{1}{b_n}$ for $n = \{1, \ldots, N_g\}$.

The standard GD scheme prescribes that, starting from a trial set of parameters, we update them against the direction of the gradient of the free energy, or, equivalently, we move towards its minimum. Instead of performing the update step after the convergence of EP algorithm (i.e. at the fixed point of $\boldsymbol{a}$ and $\boldsymbol{b}$), we can encode the learning step of the parameters within the main loop of the algorithm. Thus, at each iteration t we

update β, ρ and λ according to

$$\beta^t = \beta^{t-1} - \gamma \frac{\partial F_{EP}}{\partial \beta} \tag{8.3.5}$$

$$\rho^t = \rho^{t-1} - \gamma \frac{\partial F_{EP}}{\partial \rho} \tag{8.3.6}$$

$$\lambda^t = \lambda^{t-1} - \gamma \frac{\partial F_{EP}}{\partial \lambda} \tag{8.3.7}$$

where γ is a small constant ($\gamma \sim 10^{-5}$) that governs how fast we move towards the minimum. Notice that now convergence of EP must also depend on the refinement of these parameters. In this specific work, when have used an optimized update rule for the parameters which speed up convergence and it is called *adagrad* [75].

8.4 Results

We will show in this section some preliminary results on a 2D model of the brain, also called *ring* or *anulus*, and on a 3D model. The electric lead field matrices $\mathbf{K}$ along with the atlas $\mathbf{A}$ of the brain were kindly provided by Martínez-Montes of the Cuban Neuroscience Center.

8.4.1 Results for the ring brain

The model of the brain used for these simulations consists in 720 generators lied on a ring shaped cerebral cortex. Each generator j_n for $n \in \{1, \dots, 720\}$ has to be intended as the euclidean norm of a vectorial current density $\boldsymbol{j}_n = (j_n^x, j_n^y, j_n^z)$. The number of electrodes that provides the voltage measurements are 32. Generators are grouped together in 34 functional areas provided by a brain standard atlas of the Montreal Neurological Institute [76]. These areas are shown in figure 8.4.1.

The dictionary $\mathbf{A}$ is a matrix of dimension 720×34 where the entries of each column i corresponds to a Gaussian centered in the middle of area i and bounded in the region i; variances are designed in a way that the 99% of the more significant points of the Gaussian lie within the area i. To simulate the functioning of the brain we randomly pick K regions among 34 and we construct, for each area, a Gaussian shaped current $\boldsymbol{j}$ centered in the neighbors of the centroid of the picked areas and having variance equal to the one of the corresponding column of the dictionary. This has been done to test the "robustness" of the approach as we would like to recognize from which area of the brain the currents arrive, independently on the exact distribution of the currents within the

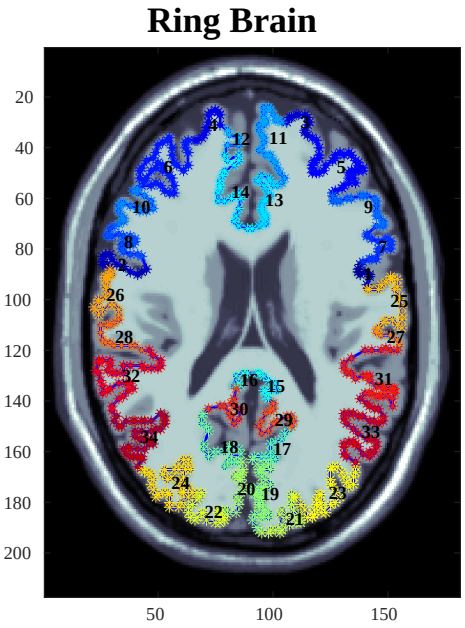


Figure 8.4.1: Ring brain partitioned in 34 functional areas

regions. Then we build the measurement vector $\boldsymbol{v}$ as $\boldsymbol{v} = \mathbf{K}\boldsymbol{j}$ and add to each component of the voltage some noise that we have considered as known. For these simulations, each component of the noise has been picked from a Gaussian distribution of zero mean and variance $\sim 10^{-2}$, i.e. $\beta \sim 10^2$. Finally we apply EP algorithm to solve the CS problem over the $\tilde{\boldsymbol{j}}$ as explained in section §8.2 along with the learning of the parameters λ and ρ outlined in section §8.3. We notice that the inference of the currents is more accurate if we let the algorithm learn these two parameters instead to fixing them to their real values.

We show some results for $K = \{1, 3\}$ in figure 8.4.2 and figure 8.4.3. The top-left plots of both figures show the vector $\boldsymbol{j}$ in the "real" space while on the top-right plots we have shown the corresponding vector $\tilde{\boldsymbol{j}}$ in the "area" space. On the bottom we report a picture of the active regions within the subdivision reported in figure 8.4.1. We can see that even though the signals, in the "real" space, do not exactly overlap in both figures the regions shown in the bottom plots exactly coincide.

8.4.2 Results for the 3D brain

These simulations concern the inference of the active regions in a 3D model of the brain. The number of generators is now 11246 while the electrodes put on the scalp are only 19. Voxels are grouped in 67 areas which correspond to functional regions; this partitioning

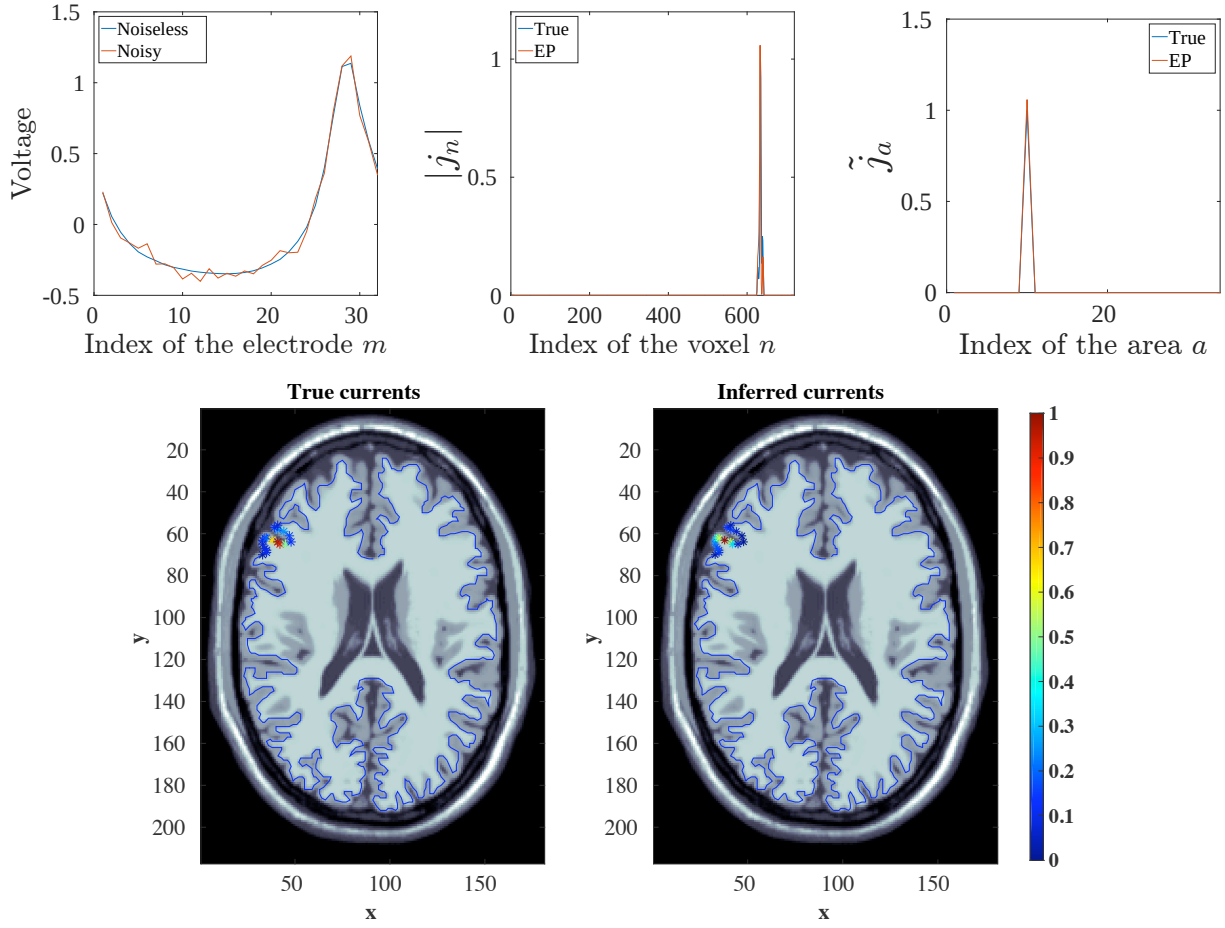


Figure 8.4.2: Result for the ring brain, one active region

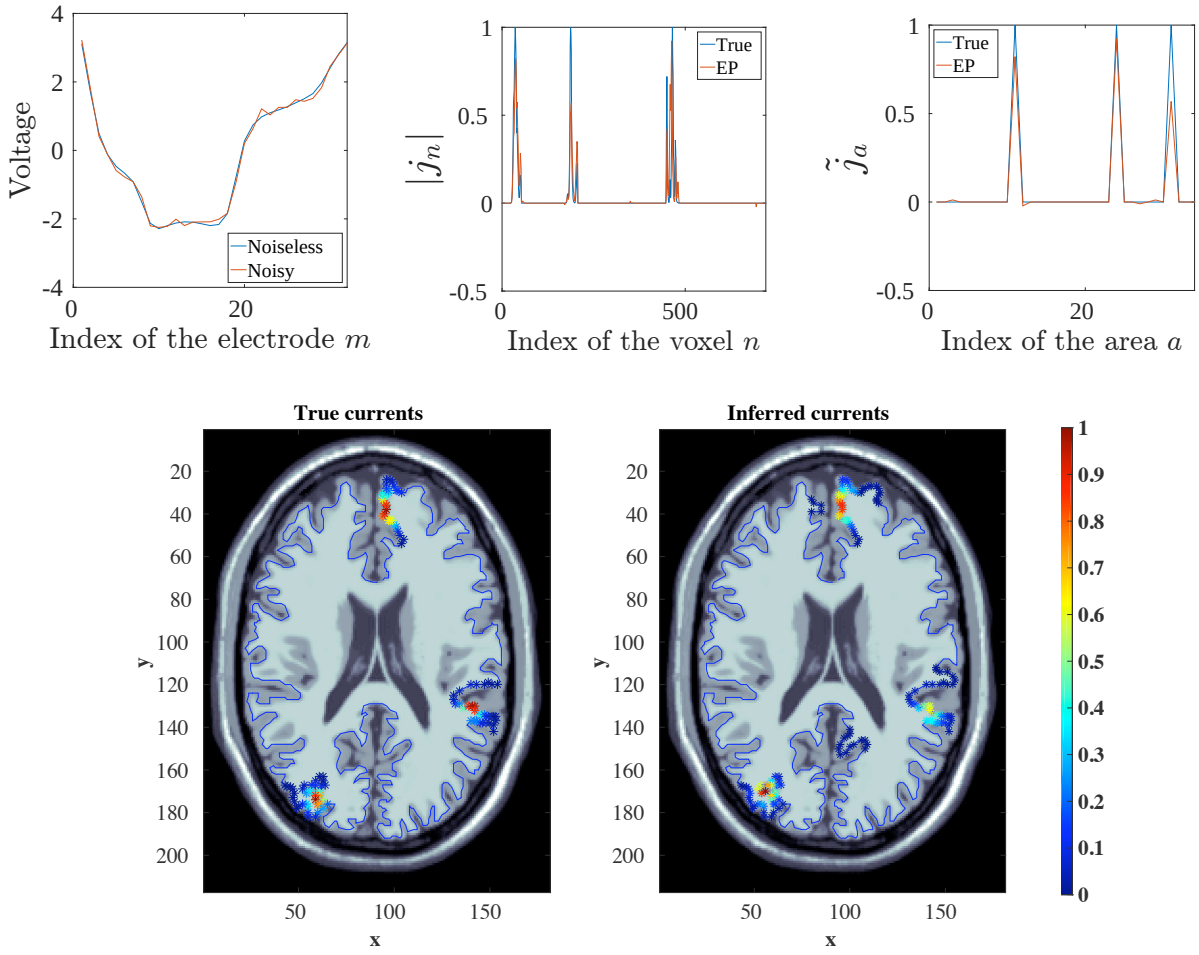


Figure 8.4.3: Result for the ring brain, three active regions

corresponds to the "Probabilistic Magnetic Resonance Imaging Atlas" developed at the Montreal Neurological Institute [77, 78] and it is pictured in figure 8.4.4. The dictionary matrix has been designed as in the case of the ring brain, that is each column i contains a Gaussian centered in the centroid of the volume of area i and properly bounded within the region. The generation process of the signals is exactly the same of the previous case. As for the anulus, we attempt to infer 1 and 3 regions and learning the parameters λ and ρ. This data has been created in a low-noise regime, i.e. imposing $\beta \sim 10^8$. Results plotted in the top figures of figure 8.4.5 and figure 8.4.6 show a very good overlap between the signals as in the real space as in the area space; this is remarked by the 3D plots of the active regions reported on the bottom figures of figure 8.4.5 and figure 8.4.6.

8.5 Discussion

In this chapter we have presented how to map the problem of inferring currents density within the brain into a compressed sensing problem of active regions in the "areas" space. The mapping has been done introducing an auxiliary matrix, the dictionary, whose design reflects our knowledge about the brain functional structure. The CS problem has then been solved by the EP algorithm introduced in chapter 6. In addition to the inference of the currents we have also shown how to learn the parameters of the model, that is, the variance of the prior λ, the sparsity in the areas space ρ, and the inverse variance of the noise β through a gradient descent method over the (approximate) free energy functional. Preliminary results concerning synthetically generated currents suggest that the method is able to predict up to 3 simultaneously activated regions as for the ring brain as for the 3D model.

In this case the learning of the parameters λ and ρ have helped to infer the signals. However, the application of the gradient descent scheme significantly slows the main loop of the EP algorithm (as it requires many iterations to let the parameters converge) even with the use of optimized version of GD, the *adagrad* method. An improved GD technique may allow the simultaneously inference of all the three parameters of the model (λ, ρ and β). In general, the computation of the main EP loop is dominated by the inversion of a matrix of size $N_a \times N_a$ where N_a is the number of areas of the brain and depends on the employed dictionary. We mention that the change of variables and the investigation in the "area" space is only possible if the matrix multiplication $\mathbf{K} \cdot \mathbf{A}$ can be performed, i.e. the number of generators N_g is not huge.

We underline that our results deeply depend on the design of the dictionary $\mathbf{A}$. In light of a future application to real EEG signals, it may be very interesting to infer also

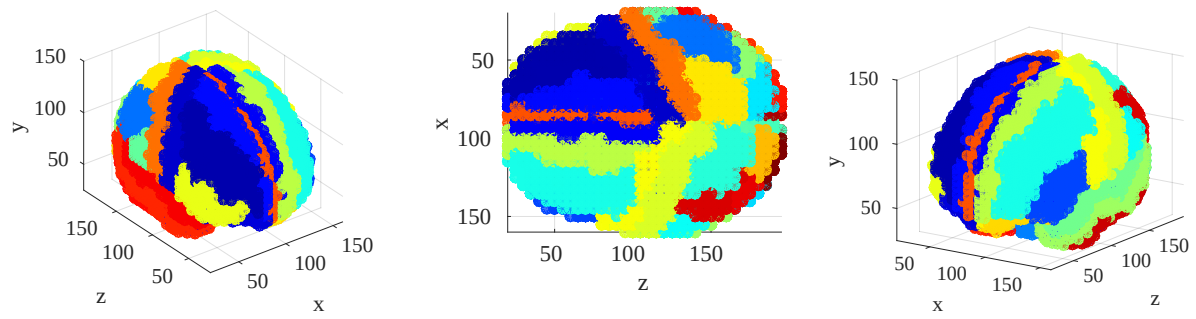


Figure 8.4.4: Partition of the brain into 67 functional areas

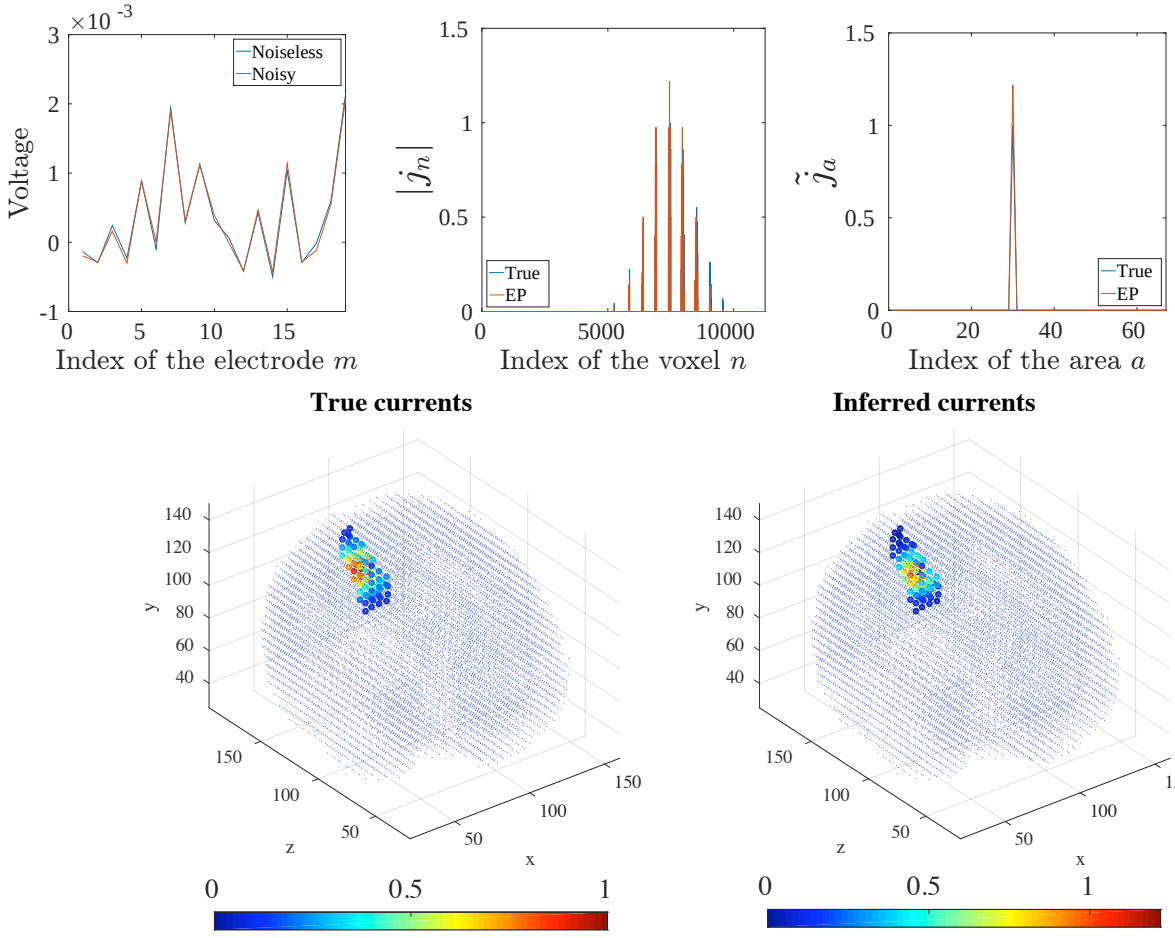


Figure 8.4.5: Result for the 3D brain, three active regions

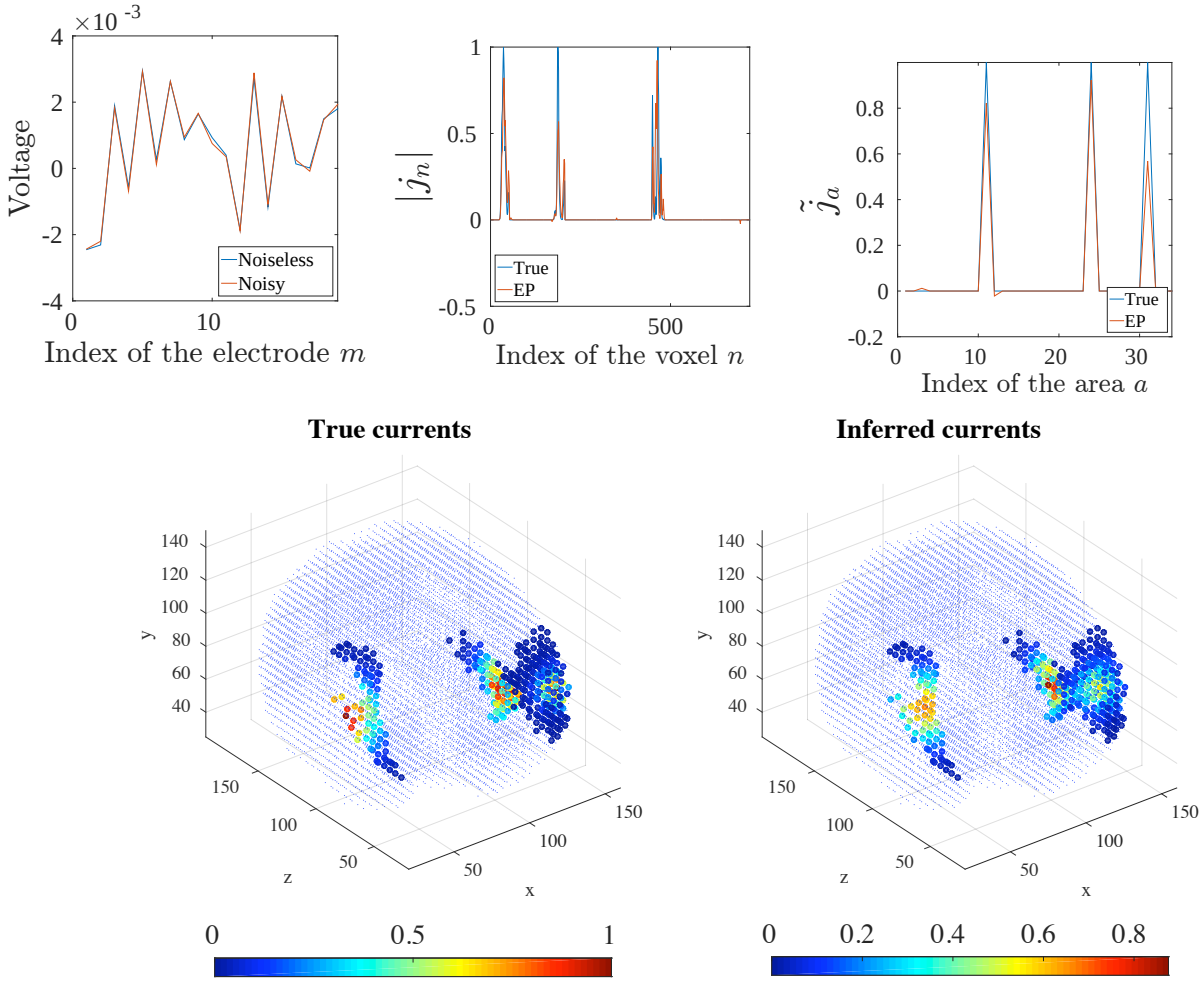


Figure 8.4.6: Result for the ring brain, three active regions

the parameters of the dictionary and so obtaining some information about the functional structure of the brain.

Chapter 9

Reconstruction of tomography images via EP

This chapter is devoted to illustrate how to treat the problem of the reconstruction of tomographic images, illustrated in section 2.2.4,within a Bayesian framework. The model we propose relies on specific choices of the prior that encompass the ability of coupling neighboring intensities/pixels within the image, of bounding their value to a pre-defined range and, eventually, of encoding *a priori* information about the image. The posterior probability that we will propose is "similar" to the one presented in [79] where, at difference to the case considered here, pixels are only modelled as binary variables. We propose several implementations of the EP algorithm able to reconstruct, as shown in the results section, a benchmark tomographic image employing very few projections. In addition, we briefly mention how to address this inference problem where the noise affecting the measurements is multiplicative, instead of additive.

All the developments presented here are part of an in progress work in collaboration with Pérez Castillo of UNAM (Universidad Autónoma de México) and his group.

9.1 Bayesian approach to tomographic imaging

As for the CS problem in section 2.2.1.1, we assume that the noise in (2.2.16) is distributed according to a Gaussian distribution of zero mean and variance $\frac{1}{\beta}$. Thus the likelihood of observing a measurements vector $\boldsymbol{y}$ can be expressed as the likelihood of observing the noise $\boldsymbol{w}$ as

$$P\left(\boldsymbol{y}|\boldsymbol{x}\right) \propto e^{-\frac{\beta}{2}(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})^T(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})} \tag{9.1.1}$$

In the prior of variables $\boldsymbol{x}$ we need to impose a spatial coupling of neighboring pixels as we assume that their intensity must change smoothly within the image. Let us introduce a prior

$$P_2(\boldsymbol{x}) \propto e^{-\frac{1}{2}\sum_{(i,j)} J(x_i - x_j)^2} = e^{-\frac{J}{2}\boldsymbol{x}^T \bar{\mathbf{L}} \boldsymbol{x}} \tag{9.1.2}$$

where J is a positive constant which governs the strength of the smoothness (or the coupling) among neighboring pixels. The matrix $\bar{\mathbf{L}}$ is the graph Laplacian defined as $\bar{\mathbf{L}} = \mathbf{G} - \mathbf{A}$ where $\mathbf{G}$ is the degree matrix and $\mathbf{A}$ the adjacency matrix associated with the lattice underlying the $L \times L$ image. Instead of using the direct definition of the Laplacian matrix, we will use the (equivalent) Laplacian defined as $\mathbf{L} = \mathbf{G}^{-\frac{1}{2}} \bar{\mathbf{L}} \mathbf{G}^{-\frac{1}{2}}$. The subscript "2" in (9.1.2) underlines the quadratic dependency on the $\boldsymbol{x}$; we mention that this kind of prior is similar to the LORETA prior usually used for the EEG inverse problem.

As explained in section §2.2.3, each pixel x_i takes value in a limited range of intensity; this knowledge is included in a second prior

$$P(\boldsymbol{x}) = \prod_i \frac{\mathbb{I}\left[x_i^{min} \leq x_i \leq x_i^{max}\right]}{x_i^{max} - x_i^{min}} = \prod_i \psi_i(x_i) \tag{9.1.3}$$

Finally, we can write the posterior of the image $\boldsymbol{x}$ given the projections $\boldsymbol{y}$ as

$$P(\boldsymbol{x}|\boldsymbol{y}) \propto e^{-\frac{\beta}{2}(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})^T(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})} e^{-\frac{J}{2}\boldsymbol{x}^T \mathbf{L} \boldsymbol{x}} \prod_i \psi_i(x_i) \tag{9.1.4}$$

As standard, one would like to make a prediction for each component x_i as

$$x_i^* = \int dx_i x_i P(x_i|\boldsymbol{y}) \tag{9.1.5}$$

but an efficient computation of these measures is intractable. In fact, the integration of several indicator functions in (9.1.4) is hard to perform. We present in section §9.2 how to derive the EP algorithm that approximates the exact posterior in (9.1.4).

9.2 EP equations

Let us define the leave-one-out distribution as

$$Q^{(i)}(\boldsymbol{x}|\boldsymbol{y}) \propto e^{-\frac{\beta}{2}(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})^T(\boldsymbol{y}-\mathbf{F}\boldsymbol{x})} e^{-\frac{J}{2}\boldsymbol{x}^T \mathbf{L} \boldsymbol{x}} \psi_i(x_i) \prod_{j \neq i} \phi_j(x_j) \tag{9.2.1}$$

where each function $\phi_j(x_j) = \frac{1}{\sqrt{2\pi b_j}} e^{-\frac{1}{2b_j}(x_j - a_j)^2}$ is aimed at approximating each exact prior $\psi_j(x_j)$. As discussed in section 3.5.1 the parameters $\boldsymbol{a}$ and $\boldsymbol{b}$ are iteratively tuned according to the update equations in (3.5.14) until we reach numeric convergence. To implement the algorithm we need to specify the mean and the variance of the tilted distribution. If we define

$$\begin{cases} \boldsymbol{\Sigma}_{(i)}^{-1} &= \beta \mathbf{F}^T \mathbf{F} + J\mathbf{L} + \mathbf{D}_{(i)} \\ \boldsymbol{\mu}^{(i)} &= \boldsymbol{\Sigma}_{(i)} \left(\beta \mathbf{F}^T \boldsymbol{y} + \mathbf{D}_{(i)} \boldsymbol{a} \right) \end{cases} \tag{9.2.2}$$

the tilted distribution reads

$$Q^{(i)}(\boldsymbol{x}|\boldsymbol{y}) \propto e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(i)}\right)^T \boldsymbol{\Sigma}_{(i)}^{-1} \left(\boldsymbol{x}-\boldsymbol{\mu}^{(i)}\right)} \psi_i(x_i) \tag{9.2.3}$$

Notice that the expression in (9.2.3) is formally equivalent to the one in (7.2.3) and thus the mean and the variance of $Q^{(i)}$ are just equal to

$$\langle x_i \rangle_{Q^{(i)}} = \mu_i + \frac{\mathcal{N}\left(\frac{x_i^{min}-\mu_i}{\sqrt{\Sigma_{ii}}}\right) - \mathcal{N}\left(\frac{x_i^{max}-\mu_i}{\sqrt{\Sigma_{ii}}}\right)}{\Phi\left(\frac{x_i^{max}-\mu_i}{\sqrt{\Sigma_{ii}}}\right) - \Phi\left(\frac{x_i^{min}-\mu_i}{\sqrt{\Sigma_{ii}}}\right)} \sqrt{\Sigma_{ii}} \tag{9.2.4}$$

$$\langle x_i^2 \rangle_{Q^{(i)}} - \langle x_i \rangle_{Q^{(i)}}^2 = \Sigma_{ii} \left[1 + \frac{\frac{x_i^{min}-\mu_i}{\Sigma_{ii}} \mathcal{N}\left(\frac{x_i^{min}-\mu_i}{\sqrt{\Sigma_{ii}}}\right) - \frac{x_i^{max}-\mu_i}{\sqrt{\Sigma_{ii}}} \mathcal{N}\left(\frac{x_i^{max}-\mu_i}{\sqrt{\Sigma_{ii}}}\right)}{\Phi\left(\frac{x_i^{max}-\mu_i}{\sqrt{\Sigma_{ii}}}\right) - \Phi\left(\frac{x_i^{min}-\mu_i}{\sqrt{\Sigma_{ii}}}\right)} + \right. \tag{9.2.5}$$

$$\left. - \left(\frac{\mathcal{N}\left(\frac{x_i^{min}-\mu_i}{\sqrt{\Sigma_{ii}}}\right) - \mathcal{N}\left(\frac{x_i^{max}-\mu_i}{\sqrt{\Sigma_{ii}}}\right)}{\Phi\left(\frac{x_i^{max}-\mu_i}{\sqrt{\Sigma_{ii}}}\right) - \Phi\left(\frac{x_i^{min}-\mu_i}{\sqrt{\Sigma_{ii}}}\right)} \right)^2 \right] \tag{9.2.6}$$

where $\mathcal{N}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ is the standard normal distribution of zero mean and unitary variance and $\Phi(x) = \int_{-\infty}^{x} \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} dy = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$ is its cumulative. Also in this case it is useful to use the expansion in (7.2.9) when $|x| \to +\infty$.

9.2.1 Estimate of the parameters

The model presented in section §9.1 introduces two parameters, β and J, that needed to be estimated. One way of determining their values is to build an Expectation Maximization scheme [73] that consists in alternating an "expectation" step, in our case an EP step for fixed values of β and J, and a "maximization" step, where the two parameters are

updated in a way that the probability of observing the data is maximized. Here we briefly derive how to perform the "maximization" step.

Let us call the probability of observing the data $\boldsymbol{y}$ and the parameters β and J as

$$P(\boldsymbol{y}, \beta, J) = \int d^N \boldsymbol{x} P(\boldsymbol{y}, \beta, J|\boldsymbol{x}) P(\boldsymbol{x}) \tag{9.2.7}$$

$$= \int \prod_i dx_i e^{-\frac{\beta}{2}(\mathbf{F}\boldsymbol{x}-\boldsymbol{y})^T(\mathbf{F}\boldsymbol{x}-\boldsymbol{y})} e^{-\frac{J}{2}\boldsymbol{x}^T\mathbf{L}\boldsymbol{x}} \prod_i \psi_i(x_i) \tag{9.2.8}$$

and let us define an energy function

$$F_{EM} = -\log P(\boldsymbol{y}, \beta, J) \tag{9.2.9}$$

The optimal values of β and J, let us call them as β^* and J^*, are the ones that minimize the energy function in (9.2.9), so we can impose that

$$\frac{\partial F_{EM}}{\partial \beta} = -\frac{1}{P(\boldsymbol{y}, \beta, J)} \frac{\partial P(\boldsymbol{y}, \beta, J)}{\partial \beta} = 0 \tag{9.2.10}$$

$$\frac{\partial F_{EM}}{\partial J} = -\frac{1}{P(\boldsymbol{y}, \beta, J)} \frac{\partial P(\boldsymbol{y}, \beta, J)}{\partial J} = 0 \tag{9.2.11}$$

If we perform the derivatives within the integral in (9.2.8), we find that

$$\beta^* = \frac{M}{\left\langle (\mathbf{F}\boldsymbol{x} - \boldsymbol{y})^T (\mathbf{F}\boldsymbol{x} - \boldsymbol{y}) \right\rangle_{P(\boldsymbol{y},\beta,J)}} \tag{9.2.12}$$

$$J^* = \frac{N}{\langle \boldsymbol{x}^T \mathbf{L} \boldsymbol{x} \rangle_{P(\boldsymbol{y},\beta,J)}} \tag{9.2.13}$$

Unfortunately, the expectations in (9.2.12) (9.2.13) cannot be efficiently computed as they require the integration of multiple error functions, but, nonetheless, as a first approximation, we can replace the computation of the first and second moments of $P(\boldsymbol{y}, \beta, J)$ with multiplications of expectations of the variables $\boldsymbol{x}$ computed through the EP algorithm. Thus the optimal values of β and J read

$$\beta^* = \frac{M}{\left(\mathbf{F}\langle \boldsymbol{x} \rangle_{Q^{EP}} - \boldsymbol{y}\right)^T \left(\mathbf{F}\langle \boldsymbol{x} \rangle_{Q^{EP}} - \boldsymbol{y}\right)} \tag{9.2.14}$$

$$J^* = \frac{N}{\langle \boldsymbol{x}^T \rangle_{Q^{EP}} \mathbf{L} \langle \boldsymbol{x} \rangle_{Q^{EP}}} \tag{9.2.15}$$

9.3 Other priors

The two priors used in section §9.1, namely the "coupling" and the "bounding" terms $P_2(\boldsymbol{x})$ and $P(\boldsymbol{x})$, are very general as they reflect the properties of tomographic, and also natural, images. We notice that, under certain conditions and for several sets of images, we can exploit much more information that can be easily encoded in the EP formalism by changing the prior $P(\boldsymbol{x})$.

Suppose that we know *a priori* that the image we aim at inferring is sparse, in the sense that a big fraction of the pixels represent a black background. In this case it is possible to introduce a prior that forces the sparsity of the image like

$$P_{sp}(\boldsymbol{x}) = \prod_i \rho\delta(x_i) + (1-\rho)\frac{\mathbb{I}\left[x_i^{min} \leq x_i \leq x_i^{max}\right]}{x_i^{max} - x_i^{min}} \tag{9.3.1}$$

where ρ is the fraction of black pixels. Notice that this prior is similar to the L_0 regularization presented in (2.2.4). Differently, for binary images, it is straightforward to impose a binary prior of the form

$$P_{bin}(\boldsymbol{x}) = \prod_i \rho\delta\left(x_i = x_i^{min}\right) + (1-\rho)\,\delta\left(x_i = x_i^{max}\right) \tag{9.3.2}$$

We do not report here the derivation of the EP update equations for these cases.

9.4 Difference variables

The main purpose of the introduction of a prior like $P_2(\boldsymbol{x})$ is to impose a coupling among neighbor pixels and, more importantly, to minimize the (discrete) second derivative of the image. It is well known, in fact, that images are usually sparse in the gradient [80, 81] and one can perform an L_1 or L_2 regularization as explained for the CS problem in equation (2.2.4). The functional forms of the corresponding priors are designed in a way that linear programming (or more generally convex optimization [82]) techniques can handle the constrained minimization problem but they cannot accommodate arbitrary distributions and, thus, the minimization of non-convex functions. This limitation is not present in the EP approximation. Our idea is to empirical study the probability distributions of the difference of pixels from images of our interest (in this particular case, a set of tomographic images of a certain portion of the body) and then design the prior according to this distribution. We also underline that, in this scenario, one can think of additionally imposing a prior over non-neighboring pixels.

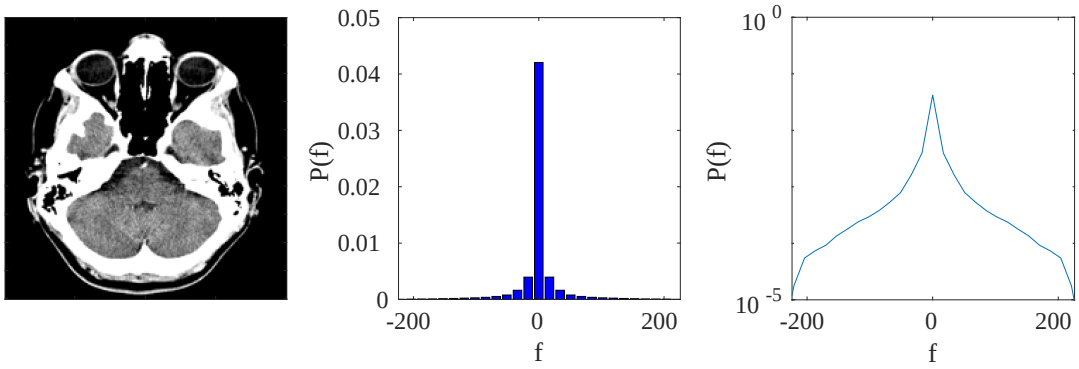


Figure 9.4.1: Left: tomographic image. Right: empirical probability distribution of the difference f

Formally, let us define as $f_{ij} = x_i - x_j$ the difference of the intensity of pixels x_i and x_j if, in the first place, are neighbors. These new variables so defined take value in the interval $f_{ij} \in [x_{min} - x_{max}, x_{max} - x_{min}]$. The trial image over which we have estimated the probability of the difference variables is shown in figure 9.4.1 and it is taken from Wikipedia. For sake of simplicity, we have computed an empirical "mean-field" probability distribution $P(f)$, for a generic difference f, that does not take into account the spatial location of the intensities. The specific localization of the differences can be easily considered by designing a set of probabilities, one for each difference-variable. Two plots of the distribution $P(f)$, differing in the scale, are reported in figure 9.4.1.

The log-scale plot in figure 9.4.1 clearly shows that this empirical distribution can be well fitted by a function of the form

$$P(f) = \rho\delta(f) + (1-\rho)\, e^{-\frac{\lambda}{2}f^2} \tag{9.4.1}$$

whose parameters ρ and λ may be fitted by the data or inferred, within the EP approximation scheme, as explained in section §8.3.

9.4.1 EP approximation

Let us introduce a Bayesian model for inferring tomographic images, known an empirical distribution for the difference-variables. The posterior probability of observing the pixels

$\boldsymbol{x}$ and the differences $\boldsymbol{f}$, known the projections $\boldsymbol{y}$, reads

$$P(\boldsymbol{x},\boldsymbol{f}|\boldsymbol{y}) \propto e^{-\frac{\beta_1}{2}(\boldsymbol{F}\boldsymbol{x}-\boldsymbol{y})^T(\boldsymbol{F}\boldsymbol{x}-\boldsymbol{y})} e^{-\frac{\beta_2}{2}\sum_{(i,j)}(x_i-x_j-f_{ij})^2} \times \\ \times \prod_i \mathbb{I}\left[x_i^{min} \le x_i \le x_i^{max}\right] \prod_{(i,j)} \left[\rho\delta(f_{ij}) + (1-\rho)\, e^{-\frac{\lambda}{2}f_{ij}^2}\right] \tag{9.4.2}$$

where the second factor of (9.4.2) imposes (for $\beta_2 \to +\infty$) the constraints among intensity-variables $\boldsymbol{x}$ and the differences; the last factor is exactly the empirical prior.

For sake of simplicity, let us call E the number of difference-variables and let us define a unique vector of unknown variables $\boldsymbol{t} = \binom{\boldsymbol{x}}{\boldsymbol{f}}$. The linear constraints involving the new unknowns can be encoded in the matrix

$$\mathbf{S}_{(M+E)\times(N+E)} = \begin{pmatrix} \sqrt{\beta_1}\boldsymbol{F} & \mathbf{0}_{M\times E} \\ \sqrt{\beta_2}\boldsymbol{R}_{E\times N} & -\sqrt{\beta_2}\mathbf{I}_{E\times E} \end{pmatrix} \tag{9.4.3}$$

where $\boldsymbol{F}$ is the matrix of the projections, $\mathbf{0}$ is the null matrix, $\boldsymbol{R}$ has zero-elements except for $R_{(i,j),i} = 1$, $R_{(i,j),j} = -1$ for every couple of pixels (i,j) and $\mathbf{I}$ is the identity matrix. The posterior probability in (9.4.2) can be rephrased as

$$P(\boldsymbol{t}) \propto e^{-\frac{1}{2}(\mathbf{S}\boldsymbol{t}-\tilde{\boldsymbol{y}})^T(\mathbf{S}\boldsymbol{t}-\tilde{\boldsymbol{y}})} \prod_{i=1}^{N} \mathbb{I}\left[x_i^{min} \le t_i \le x_i^{max}\right] \prod_{i=N+1}^{N+E} \left[\rho\delta(t_i) + (1-\rho)\, e^{-\frac{\lambda}{2}t_i^2}\right] \tag{9.4.4}$$

in which $\tilde{\boldsymbol{y}} = \binom{\boldsymbol{y}}{\mathbf{0}}^T$. We underline that the difficulty of the problem is the same as the original problem since we have added as many unknowns as equations, as it is clear from the definition of (9.4.3).

As for the problem presented in section §9.1, we can apply the EP algorithm to estimate the marginal probabilities of the variables $\boldsymbol{t}$, and then reconstruct the image from the first N components of $\boldsymbol{t}$. Since the exact prior is different as we deal with intensity or difference variables, we must specify two *tilted* distributions

$$Q^{(i)}(t_i|\boldsymbol{y}) \propto \begin{cases} e^{-\frac{1}{2}\left(\boldsymbol{t}-\boldsymbol{\mu}^{(i)}\right)^T\boldsymbol{\Sigma}_{(i)}^{-1}\left(\boldsymbol{t}-\boldsymbol{\mu}^{(i)}\right)} \mathbb{I}\left[x_i^{min} \le t_i \le x_i^{max}\right] & \text{if } i \le N \\ e^{-\frac{1}{2}\left(\boldsymbol{t}-\boldsymbol{\mu}^{(i)}\right)^T\boldsymbol{\Sigma}_{(i)}^{-1}\left(\boldsymbol{t}-\boldsymbol{\mu}^{(i)}\right)} \left[\rho\delta(t_i) + (1-\rho)\, e^{-\frac{\lambda}{2}t_i^2}\right] & \text{if } i > N \end{cases} \tag{9.4.5}$$

where

$$\begin{cases} \boldsymbol{\Sigma}_{(i)}^{-1} &= \mathbf{S}^T\mathbf{S} + \mathbf{D}_{(i)} \\ \boldsymbol{\mu}^{(i)} &= \boldsymbol{\Sigma}_{(i)}\left(\sqrt{\beta_1}\mathbf{S}^T\tilde{\boldsymbol{y}} + \mathbf{D}_{(i)}\boldsymbol{a}\right) \end{cases} \tag{9.4.6}$$

Apparently, the inversion of the matrix $\mathbf{\Sigma}_{(i)}^{-1}$ requires $\mathcal{O}\left((N+E)^3\right)$ operations that is computationally prohibitive, but, fortunately, since $\mathbf{\Sigma}_{(i)}^{-1}$ is a block matrix, it is possible to extract the diagonal blocks of $\mathbf{\Sigma}_{(i)}$ in a reduced computational cost of $\mathcal{O}(N^3)$ operations as for the intensity-variables model. In fact, if $\mathbf{\Sigma}_{(i)}^{-1}$ is

$$\mathbf{\Sigma}_{(i)}^{-1} = \begin{pmatrix} \beta_1 \mathbf{F}^T\mathbf{F} + \beta_2 \mathbf{R}^T\mathbf{R} + \mathbf{D}_1 & -\beta_2 \mathbf{R}^T \\ -\beta_2 \mathbf{R} & \beta_2 \mathbf{I} + \mathbf{D}_2 \end{pmatrix} \tag{9.4.7}$$

$$= \begin{pmatrix} \mathbf{L} & \mathbf{M} \\ \mathbf{N} & \mathbf{O} \end{pmatrix} \tag{9.4.8}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal matrices containing the values of $1/\boldsymbol{b}$ for the intensity variables and the difference variables respectively, and $\mathbf{L}$, $\mathbf{M}$, $\mathbf{N}$, $\mathbf{O}$ are auxiliary matrices, the diagonal blocks of the matrix $\mathbf{\Sigma}_{(i)}$ read

$$\mathbf{\Sigma}_{(i)} = \begin{pmatrix} \left(\mathbf{L} - \mathbf{M}\mathbf{O}^{-1}\mathbf{N}\right)^{-1} & \dots \\ \dots & \mathbf{O}^{-1}\mathbf{N}\left(\mathbf{L} - \mathbf{M}\mathbf{O}^{-1}\mathbf{N}\right)^{-1}\mathbf{M}\mathbf{O}^{-1} + \mathbf{O}^{-1} \end{pmatrix} \tag{9.4.9}$$

In (9.4.9) we must perform two inversions, $\left(\mathbf{L} - \mathbf{M}\mathbf{O}^{-1}\mathbf{N}\right)^{-1}$ and $\mathbf{O}^{-1}$, but, fortunately, computing $\mathbf{O}^{-1}$ is easy because $\mathbf{O}$ is diagonal and $\left(\mathbf{L} - \mathbf{M}\mathbf{O}^{-1}\mathbf{N}\right)$ has size $N \times N$. The non-diagonal blocks of $\mathbf{\Sigma}_{(i)}$ are not considered and thus the vector $\boldsymbol{\mu}^{(i)}$ can be determined component-wise solving the linear problem $\mathbf{\Sigma}_{(i)}^{-1}\boldsymbol{\mu}^{(i)} = \left(\sqrt{\beta_1}\mathbf{S}^T\tilde{\boldsymbol{y}} + \mathbf{D}_{(i)}\boldsymbol{a}\right)$.

To specify the EP update step, we must compute the first and second moments of (9.4.5). We only notice that they are formally equal to the ones reported in (7.2.7) (7.2.8) and in (6.1.4) (6.1.5) and are not shown here.

9.5 Multiplicative noise

In the model presented in section 2.2.4 the noise has been modelled as a Gaussian random variable that is added to each component of the measurements vector $\boldsymbol{y}$. This is not the only existing choice for dealing with noisy measurements: another idea is to introduce a multiplicative noise. We will briefly present here how to set up a Bayesian model to solve this similar inference problem to which apply, eventually, the EP algorithm. We mention that, formally, the problem is equivalent to the blind sensor calibration problem [83, 84].

As before, let us call as $\boldsymbol{w}$ the collection of Gaussian random variables affecting the noiseless projections $\hat{\boldsymbol{y}} = \mathbf{F}\boldsymbol{x}$. Formally, each linear constraint in (2.2.16) becomes

$$\boldsymbol{y} = \boldsymbol{w} \odot \mathbf{F}\boldsymbol{x} \tag{9.5.1}$$

$$= \boldsymbol{w} \odot \hat{\boldsymbol{y}} \tag{9.5.2}$$

where the symbol $\odot$ denotes the component-wise product. One way of treating this problem, is to consider each $\hat{y}_n$ as a new unknown variable to be inferred with a prior probability that can be derived from the distribution of each component of the noise $P(w_n)$. Consider the probability $P_w(w)\,dw$ for $w \in (0, +\infty)$, performing the change of variables $w = \frac{y}{\hat{y}}$, we get the distribution of the $\hat{y} \in (0, +\infty)$ as $P_{\hat{y}}(\hat{y}|y)\,d\hat{y} = P_w\left(\frac{y}{\hat{y}}\right)\frac{y}{\hat{y}^2}d\hat{y}$.

We can write the joint probability of observing $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$, given $\boldsymbol{y}$, as

$$P(\boldsymbol{x}, \hat{\boldsymbol{y}}|\boldsymbol{y}) \propto e^{-\frac{\beta}{2}(\mathbf{F}\boldsymbol{x}-\hat{\boldsymbol{y}})^T(\mathbf{F}\boldsymbol{x}-\hat{\boldsymbol{y}})} \prod_n P_{\hat{y}}(\hat{y}_n|y_n) \prod_i \psi_i(x_i) \tag{9.5.3}$$

where $\psi_i(x_i)$ is the prior over the pixel i and $\beta \to +\infty$. As for the difference-variables formalism in section §9.4, let us define a vector $\boldsymbol{t} = \binom{\boldsymbol{x}}{\hat{\boldsymbol{y}}}$ and matrix

$$\mathbf{S}_{(N+M)\times(N+M)} = \begin{pmatrix} \mathbf{F}^T\mathbf{F} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{I} \end{pmatrix} \tag{9.5.4}$$

such that (9.5.3) becomes

$$P(\boldsymbol{t}|\boldsymbol{y}) \propto e^{-\frac{\beta}{2}\boldsymbol{t}^T\mathbf{S}\boldsymbol{t}} \prod_{i \le N} \psi_i(t_i) \prod_{i > N} P_{\hat{y}}(t_i|y_{i-N}) \tag{9.5.5}$$

Also in this case it is possible to have an estimate on the marginals of $P(\boldsymbol{t}|\boldsymbol{y})$ by applying the EP approximation scheme. The details of the computation are omitted.

9.6 Preliminary results

In this section we report the results of several implementations of the EP algorithm, one for each of the presented priors,applied to a very popular benchmark tomographic image, the Shepp-Logan phantom [85], that represents a simplified picture of the human brain. The image is reshaped to size 40×40 pixels. We will compare our results to the outcomes of a convex optimization tool implemented in Matlab as the *quadprog* subroutine; in the following we will refer to this method as QuadProg. The inference problem, in this

formalism, reads

$$\boldsymbol{x}_{QuadProg} = \arg \quad \min_{\boldsymbol{x}\,:\; \mathbf{F}\boldsymbol{x} = \boldsymbol{y}} \quad \frac{1}{2}\boldsymbol{x}^T \mathbf{L} \boldsymbol{x} \tag{9.6.1}$$

Being this image in a gray-scale we will use *EPInt* (EP algorithm with flat prior on the interval as in section §9.1), *EPSparse* presented in section §9.3 and *EPdiff* where we make use of the empirical prior and the difference-variables introduced in section §9.4. The parameters ρ and λ of the difference-variables prior have been estimated by EP as explained in section §8.3.

We run EP algorithms for different values of $\alpha = \frac{M}{N}$ where $M = n_\theta \cdot 40$ is the number of measurements which define the dimension of the projection matrix $\mathbf{F}$. The design of these matrices must fit with the features of the scanning device. Often the detector can completely rotate around the object that we want to investigate and thus $\theta \in [0, 2\pi]$ or, if the rotation is limited, it can move in bounded portions of the space. To mimic the acquisition process of a beam incident on the image, we create matrices $\mathbf{F}$ where the non-zero entries of each row draws a straight line between two boundary pixels that crosses the image. Each element F_{ai} is the length of the portion of ray a passing through pixel i. The directions of the lines has been uniformly randomly chosen in the interval $[0, \pi]$ and no noise has been added to the signals.We estimate the goodness of the prediction, computing

$$Norm_2 \left(\boldsymbol{x}_{Inference} - \boldsymbol{x} \right) = \frac{1}{N} \sqrt{\sum_i \left(x^i_{Inference} - x^i \right)^2} \tag{9.6.2}$$

where $\boldsymbol{x}_{Inference}$ is the image recovered by all the EP implementations or QuadProg. We plot in figure 9.6.1 the measure $Norm_2 \left(\boldsymbol{x}_{Inference} - \boldsymbol{x} \right)$ as function of α for single runs of EP (one for each value of α) for reconstructing the Shepp-Logan image. In figure 9.6.2 we plot the original image and several reconstructed images. As shown in the plots in figure 9.6.1 and as underlined by the direct comparison of the images in figure 9.6.2, EP algorithm and QuadProg improve the predictions as we increase the number of measurements until we reach a perfect reconstruction. The behavior can be interpreted as a phase transition between a "hard" phase (in which inferring the image is very difficult) and an "easy" phase (where inference is accurate). The switch between the two regimes appears at the critical value of α, let us call it α_c, such that for $\alpha < \alpha_c$ we are in the "hard" region while for $\alpha > \alpha_c$ we jump to the "easy" region. As shown in figure 9.6.1 for this simplified single-run analysis we can estimate $\alpha_c \sim 0.57$ for *EPInt*, *EPSparse* and Quadprog. The parameters of *EPInt* and *EPSparse* have been optimized

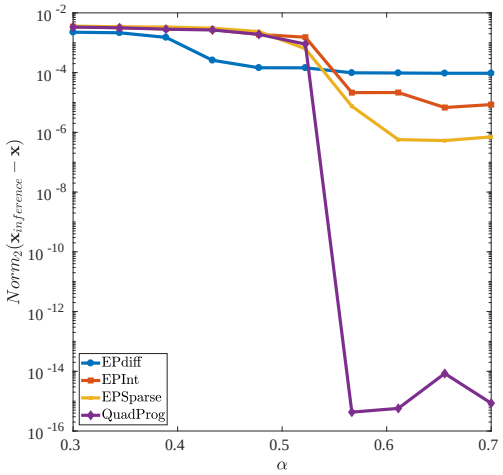


Figure 9.6.1: Reconstruction error of the Shepp-Logan image for increasing values of α

following the procedures in section 9.2.1 and section §8.3. *EPdiff*, differently, provides an almost perfect reconstruction in a very low regime, for $\alpha = 0.39$, that improves in quality for higher values of α.The performance of *EPdiff* also depend on the choice of β_1 and β_2. For this run only β_1 has been determined through the procedure in section 9.2.1; estimating also β_2 could provide a better reconstruction.

Notice that the regime in which the errors are smaller then 10^{-4} correspond to a visually perfect reconstruction of the image as it is clear from the reconstructed images in figure 9.6.2.

Discussion

As we have shown in the results above our several implementations of the EP algorithm are able to cope the problem of reconstructing tomographic images, in particular, in the case of additive noise. Results are very promising as the values of the critical α for the Shepp-Logan phantom, seems to correspond to very few projections and thus to very few measurements, matching the performance of a state-of-the-art convex optimization technique, the quadratic programming. We underline that these are preliminary results and a more rigorous analysis have to be done to estimate the goodness of the algorithms for real images and for measurements affected by noise; this is what we would like to face in future works. Moreover, the use of some auxiliary variables, the difference-variables, and the ability of EP of introducing empirical priors, open new perspectives of investigation. In fact, one could introduce other variables, linked to the intensity of the pixels through some linear transformations, and impose *ad-hoc* priors, learned from a training set, to the auxiliary variables.

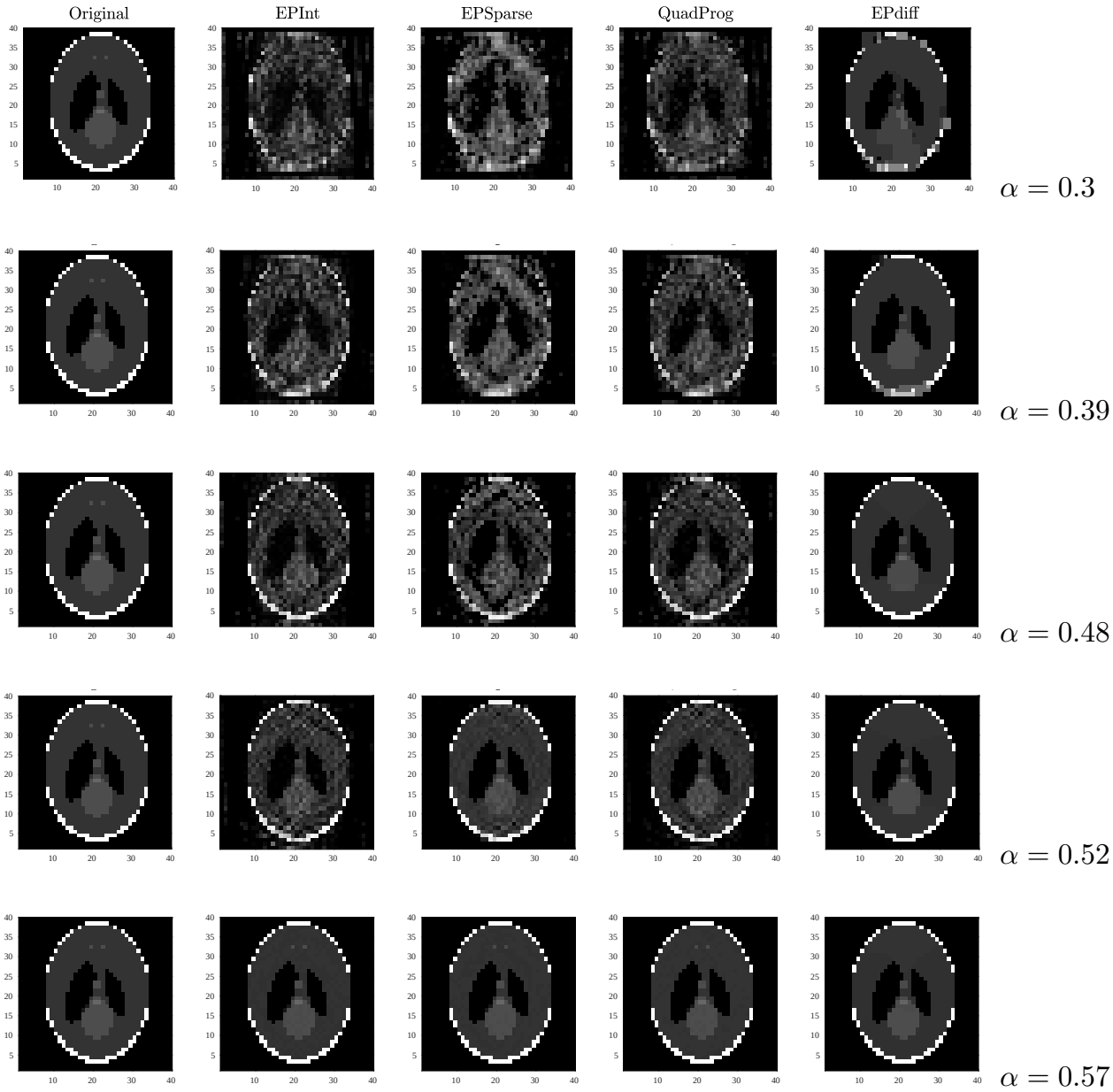


Figure 9.6.2: Reconstructed images of the Shepp-Logan image for increasing values of α

However, this method has the main bottleneck of scaling as a power 3 of the system size (see section §6.2) that is, in this case, the total number of pixels, N. For high resolution images both memorizing and inverting the matrix $\mathbf{\Sigma}^{-1}$ that has dimension $N \times N$ is barely possible. What it must be done is to seek another way of performing the inversion that, eventually, does not require to save all the data at the same time.

Appendix A

Bethe approximation

In the following we draft the main points of the Bethe approximation as in [37, 86, 1].

Let us consider the probability distribution in (3.3.11); being the beliefs marginal probabilities, they must satisfy the marginalization conditions

$$\begin{aligned} b_i(x_i) &= \sum_{x_j} b_{ij}(x_i, x_j) \\ b_{ij}(x_i, x_j) &= \sum_{\boldsymbol{x}\setminus\{x_i,x_j\}} Q(\boldsymbol{x}) \end{aligned} \tag{A.0.1}$$

As standard, we can derive a variational free energy as indicated in section §3.2 that in the case of the trial distribution in (3.3.11) and in a pairwise graphical model of probability

$$P(\boldsymbol{x}) = \frac{1}{Z}\prod_{(i,j)} \psi_{ij}(x_i, x_j) = \frac{1}{Z} e^{-\beta \sum_{(i,j)} H_{ij}(x_i,x_j)}, \tag{A.0.2}$$

takes the form of

$$\mathcal{F}_{Bethe}[Q] = U_{Bethe}[Q] - \frac{1}{\beta} S_{Bethe}[Q] \tag{A.0.3}$$

Let us specify the terms $U_{Bethe}[Q]$ and $S_{Bethe}[Q]$:

$$U_{Bethe}[Q] = -\frac{1}{\beta}\sum_{\boldsymbol{x}}\sum_{(i,j)} Q(\boldsymbol{x}) \ln \psi_{ij}(x_i, x_j) \tag{A.0.4}$$

$$= -\frac{1}{\beta}\sum_{(i,j)}\sum_{x_i,x_j} b_{ij}(x_i, x_j) \ln \psi_{ij}(x_i, x_j) \tag{A.0.5}$$

$$S_{Bethe}[Q] = -\sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \ln \left[\prod_{(i,j)} \frac{b_{ij}(x_i, x_j)}{b_i(x_i)\, b_j(x_j)} \prod_i b_i(x_i) \right] \tag{A.0.6}$$

$$= -\sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \left[\sum_{(i,j)} \ln b_{ij}(x_i, x_j) - \sum_i (d_i - 1) \ln b_i(x_i) \right] \tag{A.0.7}$$

$$= -\sum_{(i,j)} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln b_{ij}(x_i, x_j) + \sum_i (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i) \tag{A.0.8}$$

where d_i is the degree of node i. The Bethe free energy will read:

$$\mathcal{F}_{Bethe}[Q] = -\frac{1}{\beta} \left[\sum_{(i,j)} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln \psi_{ij}(x_i, x_j) - \sum_{(i,j)} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \ln b_{ij}(x_i, x_j) + \right.$$
$$\left. + \sum_i (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i) \right] \tag{A.0.9}$$

Additionally we need to ensure that: (i) one and two-nodes belief are normalized and (ii) the condition expressed in (A.0.1) is satisfied. According to the Lagrangian formalism, we add as many Lagrange multipliers as the constraints to the free energy in (A.0.9):

$$\mathcal{F}[Q] = \mathcal{F}_{Bethe}[Q] +$$
$$+ \sum_i \lambda_i \left[\sum_{x_i} b_i(x_i) - 1 \right] + \sum_i \sum_{j \in \partial i} \sum_{x_i} \lambda_{(i,j)\to i}(x_i) \left[\sum_{x_j} b_{ij}(x_i, x_j) - b_i(x_i) \right]$$

Now if we set the derivatives of $\mathcal{F}$ with respect to the Lagrangian multipliers to be zero, we encounter the marginalization and normalization constraints. Instead, imposing the derivatives with respect to the beliefs to zero, we find

$$\begin{cases} 0 &= \frac{\partial \mathcal{F}}{\partial b_i(x_i)} = -(d_i - 1)[\ln b_i(x_i) + 1] + \lambda_i - \sum_{j \in \partial i} \lambda_{(i,j)\to i}(x_i) \\ 0 &= \frac{\partial \mathcal{F}}{\partial b_{ij}(x_i, x_j)} = -\ln \psi_{ij}(x_i, x_j) + [\ln b_{ij}(x_i, x_j) + 1] - \lambda_{(i,j)\to j}(x_j) - \lambda_{(i,j)\to i}(x_i) \end{cases} \tag{A.0.10}$$

Thus the beliefs at the stationary point are given by

$$\begin{cases} b_i^*(x_i) & \propto e^{\frac{1}{d_i - 1} \left[\lambda_i - \sum_{j \in \partial i} \lambda_{(i,j)\to i}(x_i) \right]} \\ b_{ij}^*(x_i, x_j) & \propto \psi_{ij}(x_i, x_j)\, e^{\left[-\lambda_{(i,j)\to i}(x_i) - \lambda_{(i,j)\to j}(x_j) \right]} \end{cases} \tag{A.0.11}$$

where the Lagrange multipliers must be chosen in a way that the marginalization condition of the beliefs is satisfied (normalizing (A.0.11) one finds λ_i as a function of the other Lagrange multipliers). We left to show that this set of Lagrange multipliers is in one-to-one correspondence to Belief Propagation fixed points.

Let us introduce the messages

$$m_{i\to(i,j)}\left(x_i\right) \propto e^{-\lambda_{(i,j)\to i}(x_i)} \qquad m_{(i,j)\to i}\left(x_i\right) \propto \sum_{x_j} \psi_{ij}\left(x_i, x_j\right) e^{-\lambda_{(i,j)\to j}(x_j)} \tag{A.0.12}$$

that surely satisfy

$$m_{(i,j)\to i}\left(x_i\right) \propto \sum_{x_j} \psi_{ij}\left(x_i, x_j\right) m_{j\to(i,j)}\left(x_j\right) \tag{A.0.13}$$

If we combine the second equation of (A.0.11) to (A.0.12) we obtain

$$\sum_{x_j} b_{ij}\left(x_i, x_j\right) \propto m_{i\to(i,j)}\left(x_i\right) m_{(i,j)\to i}\left(x_i\right) \tag{A.0.14}$$

while using the first equation of (A.0.11) together with (A.0.12) we get

$$b_i\left(x_i\right) \propto \prod_{j\in\partial i} \left[m_{i\to(i,j)}\right]^{\frac{1}{d_i-1}} \tag{A.0.15}$$

Replacing the beliefs in (A.0.14) and in (A.0.15) within the first equation of (A.0.1) we find the following relation for the messages

$$\prod_{k\in\partial i} \left[m_{i\to(i,k)}\left(x_i\right)\right]^{\frac{1}{d_i-1}} \propto m_{i\to(i,j)}\left(x_i\right) m_{(i,j)\to i}\left(x_i\right) \tag{A.0.16}$$

Multiplying both the left-hand side and the right-hand side of (A.0.16) by $\prod_{j\in\partial i\setminus k}$ and for $\psi_{ij} > 0$, we obtain

$$m_{i\to(i,k)}\left(x_i\right) \propto \prod_{j\in\partial i\setminus k} m_{(i,j)\to i}\left(x_i\right) \tag{A.0.17}$$

Notice that equations (A.0.13) and (A.0.17) represent the fixed point equations of the BP algorithm. Thus, given a set of Lagrange multipliers at the stationary points of the Bethe free energy functional, we can define, as in (A.0.12), a set of messages that satisfy the fixed point BP equations. On the other hand, from the fixed point of BP, we can invert the relations in (A.0.12) and together with the marginalization condition, obtain a the Lagrange multipliers at the stationary point of the Bethe free energy.

Appendix B

BP equations for the CS problem

From the definition of $m_{n\to a}(x_n)$ and the final expression of message $m_{a\to n}(x_n)$, we obtain

$$m_{n\to a}(x_n) \propto \frac{1}{\tilde{Z}_{n\to a}}\left[(1-\rho)\,\delta(x_n) + \rho\sqrt{\frac{1}{2\pi\lambda}}e^{-\frac{x_n^2}{2\lambda}}\right] e^{-\frac{x_n^2}{2}\sum_{b\neq a}A_{b\to n} + x_n\sum_{b\neq a}B_{b\to n}} \quad \text{(B.0.1)}$$

where the normalization factor is computed as

$$\tilde{Z}_{n\to a} = (1-\rho) + \rho\sqrt{\frac{1}{1+\lambda\sum_{b\neq a}A_{b\to n}}}e^{\frac{\left(\lambda\sum_{b\neq a}B_{b\to n}\right)^2}{2\left(1+\lambda\sum_{b\neq a}A_{b\to n}\right)}} \quad \text{(B.0.2)}$$

We can now explicit the first and second moments of the approximate Gaussian message as

$$\begin{aligned}
\mu_{n\to a} &= \int dx_n x_n m_{n\to a}(x_n) \\
&= \frac{1}{\tilde{Z}_{n\to a}}\rho\sqrt{\frac{1}{2\pi\lambda}}e^{\frac{\left(\sum_{b\neq a}B_{b\to n}\right)^2}{2\left(\lambda+\sum_{b\neq a}A_{b\to n}\right)}}\int dx_n x_n e^{-\frac{\left(1+\lambda\sum_{b\neq a}A_{b\to n}\right)}{2\lambda}\left(x_n-\frac{\lambda\sum_{b\neq a}B_{b\to a}}{1+\lambda\sum_{b\neq a}A_{b\to n}}\right)^2} \\
&= \frac{1}{\tilde{Z}_{n\to a}}\rho\sqrt{\frac{1}{1+\lambda\sum_{b\neq a}A_{b\to n}}}e^{\frac{\left(\lambda\sum_{b\neq a}B_{b\to n}\right)^2}{2\left(1+\lambda\sum_{b\neq a}A_{b\to n}\right)}}\frac{\lambda\sum_{b\neq a}B_{b\to n}}{1+\lambda\sum_{b\neq a}A_{b\to n}}
\end{aligned} \quad \text{(B.0.3)}$$

$$\sigma_{n\to a}^2 = \frac{1}{\tilde{Z}_{n\to a}} \int dx_n x_n^2 \left[(1-\rho)\,\delta(x_n) + \rho\sqrt{\frac{1}{2\pi\lambda}} e^{-\frac{1}{2\lambda}x_n^2} \right] e^{-\frac{1}{2}x_n^2 \sum_{b\neq a} A_{b\to n} + x_n \sum_{b\neq a} B_{b\to n}} - \mu_{n\to a}^2$$

$$= \frac{1}{\tilde{Z}_{n\to a}} \rho \sqrt{\frac{1}{1+\lambda\sum_{b\neq a} A_{b\to n}}} e^{\frac{\left(\lambda\sum_{b\neq a} B_{b\to n}\right)^2}{2\left(1+\lambda\sum_{b\neq a} A_{b\to n}\right)}} \left[\frac{\lambda}{\left(1+\lambda\sum_{b\neq a} A_{b\to n}\right)} + \left(\frac{\lambda\sum_{b\neq a} B_{b\to n}}{1+\lambda\sum_{b\neq a} A_{b\to n}}\right)^2 + \right.$$

$$\left. - \frac{1}{\tilde{Z}_{n\to a}} \rho \sqrt{\frac{1}{1+\lambda\sum_{b\neq a} A_{b\to n}}} e^{\frac{\left(\lambda\sum_{b\neq a} B_{b\to n}\right)^2}{2\left(1+\lambda\sum_{b\neq a} A_{b\to n}\right)}} \left(\frac{\lambda\sum_{b\neq a} B_{b\to n}}{1+\lambda\sum_{b\neq a} A_{b\to n}}\right)^2 \right] \tag{B.0.4}$$

If one defines

$$W_{n\to a} = \rho \sqrt{\frac{1}{1+\lambda\sum_{b\neq a} A_{b\to n}}} e^{\frac{\left(\lambda\sum_{b\neq a} B_{b\to n}\right)^2}{2\left(1+\lambda\sum_{b\neq a} A_{b\to n}\right)}} \tag{B.0.5}$$

we get

$$\mu_{n\to a} = \frac{1}{\tilde{Z}_{n\to a}} W_{n\to a} \frac{\lambda\sum_{b\neq a} B_{b\to n}}{1+\lambda\sum_{b\neq a} A_{b\to n}^t}$$

$$\sigma_{n\to a}^2 = \frac{1}{\tilde{Z}_{n\to a}} W_{n\to a} \left[\left(\frac{\lambda\sum_{b\neq a} B_{b\to n}}{1+\lambda\sum_{b\neq a} A_{b\to n}}\right)^2 \left(1 - \frac{W_{n\to a}}{\tilde{Z}_{n\to a}}\right) + \frac{\lambda}{\left(1+\lambda\sum_{b\neq a} A_{b\to n}\right)} \right] \tag{B.0.6}$$

Appendix C

Standard derivation of EP algorithm

Minka has presented EP algorithm in [41] as a powerful tool to approximate posterior probabilities. The main idea behind his approach is to improve an *online learning* algorithm, the Assumed Density Filtering (ADF) [87–89], which sequentially modifies the posterior including, one by one, any intractable term coming from new observations of data. EP can be seen in this prospective as the *batch* version of ADF where the intractable terms iteratively modify the posterior until we reach a convergence condition. Differently from the approximation described in section 3.5.1, Minka's derivation and ADF deal with a posterior distribution in which the Gaussian approximation is performed on an intractable likelihood. First, let us briefly introduce the ADF algorithm.

Suppose of observing a set of data $D = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ from which we want to infer a hidden vector $\boldsymbol{x}$. Let us specify the likelihood and the prior that, via Bayes' theorem, enter in the joint distribution of $\boldsymbol{x}$, that is $P(\boldsymbol{x}|D)$. Since D is a collection of N independent observations we assume $P(D|\boldsymbol{x}) = \prod_{a=1}^{N} P(\boldsymbol{y}_a|\boldsymbol{x})$. Even if the prior $P(\boldsymbol{x})$ can be tractable, the computation of the posterior $P(\mathbf{x}|D)$ can be hard to treat depending on the form of the likelihood functions.

The idea of Lauritzen et all [89] is to sequentially modify a trial probability $Q(\boldsymbol{x})$ as we observe a new set of data. First, we choose to which approximate family $Q(\boldsymbol{x})$ must belong; if the prior is a tractable distribution one can think of choosing an objective distribution of the same family. For sake of simplicity we consider the latter case and we initialize $Q(\boldsymbol{x}) = P(\boldsymbol{x})$.

Let us consider the set of data $\boldsymbol{y}_1$ and let us modify the posterior accordingly to the first measures:

$$P^{a=1}(\boldsymbol{y}_1|D) = \frac{Q(\boldsymbol{x})\,P(\boldsymbol{y}_1|\boldsymbol{x})}{\int_{\mathbf{x}} Q(\boldsymbol{x})\,P(\boldsymbol{y}_1|\boldsymbol{x})} \tag{C.0.1}$$

In order to compute a *new* approximate distribution $Q^{new}(\boldsymbol{x})$ as similar as possible to $P^{a=1}(\boldsymbol{x}|D)$ we can minimize $D_{KL}[P^{a=1}\|Q^{new}]$ or, equivalently, modify the parameters of Q^{new} to let the first and second moments of $P^{a=1}$ and Q^{new} match. We then update $Q = Q^{new}$ and we *propagate* these moments in the sense that if now observe $\boldsymbol{y}_2$ and we build the partial posterior $P^{a=2}(\boldsymbol{x}|D) \propto Q(\boldsymbol{x})\,P(\boldsymbol{y}_2|\boldsymbol{x})$, the information regarding $\boldsymbol{y}_1$ is absorbed in the expectations of $Q(\boldsymbol{x})$. We repeat the same procedure as before updating Q until we observe all N measures. The main weakness of this algorithm is that is strongly dependent on the order on which we observe the data. This drawback is successfully overcame by EP that can be seen, in this perspective, as an iteratively refinement of the approximate probability $Q(\boldsymbol{x})$ introduced by ADF.

Appendix D

Moments matching condition

Let us now rewrite the probability distributions (3.5.3) and (3.5.4) pointing out the co-factor (3.5.10) and making explicit the dependency of the normalization factors with respect to the parameters a_n, b_n:

$$Q^{(n)}(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{\tilde{Z}_{Q^{(n)}}} e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)^T \boldsymbol{\Sigma}_{(n)}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)} \psi_n(x_n) \tag{D.0.1}$$

$$Q(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{\tilde{Z}_Q(a_n, b_n)} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}^{(n)})^T \boldsymbol{\Sigma}_{(n)}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(n)})} e^{-\frac{(x_n-a_n)^2}{2b_n}} \tag{D.0.2}$$

where the partition functions take the values

$$\tilde{Z}_{Q^{(n)}} = \int d^N \boldsymbol{x} e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)^T \boldsymbol{\Sigma}_{(n)}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)} \psi_n(x_n) \tag{D.0.3}$$

$$\tilde{Z}_Q(a_n, b_n) = \int d^N \boldsymbol{x} e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)^T \boldsymbol{\Sigma}_{(n)}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)} e^{-\frac{(x_n-a_n)^2}{2b_n}} \tag{D.0.4}$$

Let us compute $D_{KL}\left[Q^{(n)} \| Q\right]$ step-by-step

$$\begin{aligned} D_{KL}\left[Q^{(n)} \| Q\right] &= \int d^N \boldsymbol{x} Q^{(n)}(\boldsymbol{x}|\boldsymbol{y}) \log\left[\frac{\psi_n(x_n)\tilde{Z}_Q(a_n, b_n)}{\phi_n(x_n)\tilde{Z}_{Q^{(n)}}}\right] \\ &= \int d^N \boldsymbol{x} Q^{(n)}(\boldsymbol{x}|\boldsymbol{y}) \left[\frac{(x_n - a_n)^2}{2b_n} + \log \tilde{Z}_Q(a_n, b_n)\right] + const \\ &= \frac{\langle (x_n - a_n)^2 \rangle_{Q^{(n)}}}{2b_n} + \log \tilde{Z}_Q(a_n, b_n) + const \end{aligned} \tag{D.0.5}$$

where *const* contains all the terms not depending neither on a_n nor on b_n. Let us minimize $D_{KL}\left[Q^{(n)}\|Q\right]$ with respect to a_n and b_n:

$$\frac{\partial D_{KL}\left[Q^{(n)}\|Q\right]}{\partial a_n} = \frac{-\langle x_n\rangle_{Q^{(n)}} + a_n}{b_n} + \frac{1}{\tilde{Z}_Q}\frac{\partial \tilde{Z}_Q}{\partial a_n} \tag{D.0.6}$$

$$\frac{\partial D_{KL}\left[Q^{(n)}\|Q\right]}{\partial b_n} = -\frac{\langle (x_n - a_n)^{\,2}\rangle_{Q^{(n)}}}{2b_n^2} + \frac{1}{\tilde{Z}_Q}\frac{\partial \tilde{Z}_Q}{\partial b_n} \tag{D.0.7}$$

Since we can move the derivative inside the integration in $\frac{\partial \tilde{Z}_Q}{\partial a_n}$ and in $\frac{\partial \tilde{Z}_Q}{\partial b_n}$ we get:

$$\begin{aligned}\frac{1}{\tilde{Z}_Q}\frac{\partial \tilde{Z}_Q}{\partial a_n} &= \frac{1}{\tilde{Z}_Q}\int d^N\boldsymbol{x} e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)^T\boldsymbol{\Sigma}_{(n)}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)} e^{-\frac{(x_n-a_n)^2}{2b_n}}\frac{(x_n-a_n)}{b_n} \\ &= \langle\frac{x_n-a_n}{b_n}\rangle_Q\end{aligned}$$

$$\begin{aligned}\frac{1}{\tilde{Z}_Q}\frac{\partial \tilde{Z}_Q}{\partial b_n} &= \frac{1}{\tilde{Z}_Q}\int d^N\boldsymbol{x} e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)^T\boldsymbol{\Sigma}_{(n)}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}^{(n)}\right)} e^{-\frac{(\nu_n-a_n)^2}{2b_n}}\frac{(x_n-a_n)^2}{2b_n^2} \\ &= \frac{\langle(x_n-a_n)^2\rangle_Q}{2b_n^2}\end{aligned}$$

Setting the derivatives in (D.0.7) to 0 and assuming $b_n \neq 0$ we finally get

$$\begin{cases} 0 &= \frac{-\langle x_n\rangle_{Q^{(n)}}+a_n}{b_n} + \frac{\langle x_n\rangle_Q - a_n}{b_n} \\ 0 &= -\frac{\langle(x_n-a_n)^2\rangle_{Q^{(n)}}}{2b_n^2} + \frac{\langle(x_n-a_n)^2\rangle_Q}{2b_n^2}\end{cases} \tag{D.0.8}$$

$$\begin{cases}\langle x_n\rangle_{Q^{(n)}} &= \langle x_n\rangle_Q \\ \langle x_n^2\rangle_{Q^{(n)}} &= \langle x_n^2\rangle_Q\end{cases} \tag{D.0.9}$$

and thus the moment matching condition in (3.5.8) turns out to be equivalent to the Kullback-Leibler divergence minimization condition.

Appendix E

EP free energy functional

Let us define $Q_{EP}(\boldsymbol{x})$ the normalized probability density approximating an exact posterior $P(\boldsymbol{x}|\boldsymbol{y})$. The partition function can be written as

$$Z_{EP} = Z_{EP} \int d^N \boldsymbol{x} Q_{EP}(\boldsymbol{x}) \tag{E.0.1}$$

The distribution Q_{EP} is exactly the distribution in (3.5.4) at the fixed point of the algorithm. This is the product of $N+1$ factors $\{\phi_0, \phi_1, \ldots, \phi_N\}$ constituting with their own unknown normalization factors $\{Z_0, Z_1, \ldots Z_{N+1}\}$. In particular, $\phi_0(\boldsymbol{x}) = P(\boldsymbol{y}|\boldsymbol{x})$, $Z_0 = \int d^M \boldsymbol{y} P(\boldsymbol{y}|\boldsymbol{x})$ whereas the functions $\phi_n(x_n)$, $n = \{1, \ldots, N\}$, are Gaussian distributions that at convergence satisfy the moment-matching constraint in (3.5.8). Thus

$$\begin{aligned} Z_{EP} &= \int d^N \boldsymbol{x} \prod_n Z_n \phi_n(\boldsymbol{x}) && \text{(E.0.2)} \\ &= \left(\prod_n Z_n\right) \int d^N \boldsymbol{x} \phi_0(\boldsymbol{x}) \prod_{n>0} \phi_n(x_n) && \text{(E.0.3)} \\ &= \left(\prod_n Z_n\right) Z_Q && \text{(E.0.4)} \end{aligned}$$

As stated in appendix D, the parameters of the functions $\phi_n(\nu_n)$, $n = \{1, \ldots, N\}$ minimize of the Kullback-Leibler divergence in (3.5.6) or, using a shorter notation, the Gaussian distributions satisfy

$$\arg\min_{\phi_n} D_{KL}\left[Q^{(n)}(\boldsymbol{\nu}|\boldsymbol{b}) \,||Q(\boldsymbol{\nu}|\boldsymbol{b})\right] \quad n = \{1, \ldots, N\} \tag{E.0.5}$$

Equivalently we aim at determining each partial partition functions Z_n as

$$\arg\min_{Z_n} D_{KL}\left[Q^{(n)}(\boldsymbol{\nu}|\boldsymbol{b}) \,||Q(\boldsymbol{\nu}|\boldsymbol{b})\right] \quad n = \{1, \ldots, N\} \tag{E.0.6}$$

If we set to zero the derivatives with respect to Z_n we finally obtain an expression of the partial partition function:

$$Z_n = \frac{1}{Z_Q} \int d^N \boldsymbol{x} \prod_m \phi_m (x_m) \left[\frac{\psi_n (\nu_n)}{\phi_n (\nu_n)} \right] \tag{E.0.7}$$

$$= \left\langle \frac{\psi_n (\nu_n)}{\phi_n (\nu_n)} \right\rangle_Q \tag{E.0.8}$$

Now we can define a free energy functional $F_{EP} = -\log Z_{EP}$:

$$F_{EP} = -\log Z_Q - \sum_n \log Z_n \tag{E.0.9}$$

Bibliography

[1] Marc Mézard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, Inc., New York, NY, USA, 2009.

[2] R. K. Pathria and Paul D. Beale. *Statistical Mechanics*. Academic Press, April 2011.

[3] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[4] Allon Percus, Gabriel Istrate, and Cristopher Moore. *Computational Complexity and Statistical Physics*. OUP USA, 2006. Google-Books-ID: 4YD6AxV95zEC.

[5] Stephen A. Cook. The Complexity of Theorem-proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[6] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[7] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh Virkumar Vazirani. Algorithms. 2016.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[9] Lenka Zdeborová and Florent Krzakala. Statistical physics of inference: Thresholds and algorithms. *Advances in Physics*, 65(5):453–552, September 2016. arXiv: 1511.02476.

[10] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[11] E. T. Jaynes. Information Theory and Statistical Mechanics. *Physical Review*, 106(4):620–630, May 1957.

[12] Hidetoshi Nishimori. *Statistical Physics of Spin Glasses and Information Processing*. Oxford University Press, July 2001.

[13] Ivana Ljubić, René Weiskircher, Ulrich Pferschy, Gunnar W. Klau, Petra Mutzel, and Matteo Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical programming*, 105(2-3):427–449, 2006.

[14] Nurcan Tuncbag, Alfredo Braunstein, Andrea Pagnani, Shao-ShanCarol Huang, Jennifer Chayes, Christian Borgs, Riccardo Zecchina, and Ernest Fraenkel. Simultaneous reconstruction of multiple signaling pathways via the prize-collecting steiner forest problem. In Benny Chor, editor, *Research in Computational Molecular Biology*, volume 7262 of *Lecture Notes in Computer Science*, pages 287–301. Springer Berlin Heidelberg, 2012. Cited by 0002.

[15] M. Bailly-Bechet, C. Borgs, A. Braunstein, J. Chayes, A. Dagkessamanskaia, J.-M. François, and R. Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *Proceedings of the National Academy of Sciences*, 108(2):882–887, January 2011.

[16] Nam-Dung Hoáng and Thorsten Koch. Steiner tree packing revisited. *Mathematical Methods of Operations Research*, 76(1):95–123, 2012.

[17] Ivana Ljubić, René Weiskircher, Ulrich Pferschy, Gunnar W. Klau, Petra Mutzel, and Matteo Fischetti. An Algorithmic Framework for the Exact Solution of the Prize-Collecting Steiner Tree Problem. *Math. Program.*, 105(2-3):427–449, October 2005.

[18] Marcus Poggi de Aragao, Eduardo Uchoa, and Renato F. Werneck. Dual heuristics on the exact solution of large Steiner problems. *Electronic Notes in Discrete Mathematics*, 7:150–153, 2001.

[19] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '92, pages 307–316, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics.

[20] M. Bayati, C. Borgs, A. Braunstein, J. Chayes, A. Ramezanpour, and R. Zecchina. Statistical mechanics of steiner trees. *Physical Review Letters*, 101(3):037208, July 2008.

[21] Indaco Biazzo, Alfredo Braunstein, and Riccardo Zecchina. Performance of a cavity-method-based algorithm for the prize-collecting steiner tree problem on graphs. *Phys. Rev. E*, 86:026706, August 2012.

[22] E.J. Candes and M.B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008.

[23] M. F. Duarte, M. A. Davenport, D. Takbar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-Pixel Imaging via Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, March 2008.

[24] J. Bobin, J. L. Starck, and R. Ottensamer. Compressed Sensing in Astronomy. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):718–726, October 2008.

[25] Vincent Studer, Jérome Bobin, Makhlad Chahid, Hamed Shams Mousavi, Emmanuel Candes, and Maxime Dahan. Compressive fluorescence microscopy for biological and hyperspectral imaging. *Proceedings of the National Academy of Sciences*, 109(26):E1679–E1687, June 2012.

[26] Florent Krzakala, Marc Mézard, Francois Sausset, Yifan Sun, and Lenka Zdeborová. Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(08):P08009, 2012.

[27] Trevor Park and George Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, June 2008.

[28] D. L. Donoho. Compressed Sensing. *IEEE Trans. Inf. Theor.*, 52(4):1289–1306, April 2006.

[29] David L. Donoho, Arian Maleki, and Andrea Montanari. Message passing algorithms for compressed sensing: I. motivation and construction. In *Information Theory (ITW 2010, Cairo), 2010 IEEE Information Theory Workshop on*, pages 1–5. IEEE, 2010.

[30] Y. Kabashima, T. Wadayama, and T. Tanaka. A typical reconstruction limit for compressed sensing based on L p -norm minimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(09):L09003, 2009.

[31] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 4th edition, 2002.

[32] David L Nelson, Albert L Lehninger, and Michael M Cox. *Lehninger principles of biochemistry*. Macmillan, 2008.

[33] A. De Martino and E. Marinari. The solution space of metabolic networks: Producibility, robustness and fluctuations. In *Journal of Physics: Conference Series*, volume 233, page 012019. IOP Publishing, 2010.

[34] Deirel Paz-Linares, Mayrim Vega-Hernández, Pedro A. Rojas-López, Pedro A. Valdés-Sosa, and Eduardo Martínez-Montes. Empirical bayes formulation of the elastic net and mixed-norm models: application to the eeg inverse problem. *arXiv preprint arXiv:1601.06749*, 2016.

[35] A. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, January 2001.

[36] Nadiya Gubareni. Algebraic Algorithms for Image Tomographic Reconstruction from Incomplete Projection Data. In Safeeullah Soomro, editor, *Engineering the Computer Science and IT*. InTech, October 2009.

[37] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, July 2005.

[38] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.

[39] H. A. Bethe. Statistical Theory of Superlattices. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 150(871):552–575, 1935.

[40] S. Rangan. Estimation with random linear mixing, belief propagation and compressed sensing. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2010.

[41] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[42] Manfred Opper and Ole Winther. Expectation consistent approximate inference. *The Journal of Machine Learning Research*, 6:2177–2204, 2005.

[43] Manfred Opper and Ole Winther. Adaptive and self-averaging Thouless-Anderson-Palmer mean-field theory for probabilistic modeling. *Physical Review E*, 64(5):056131, October 2001.

[44] M. Opper and O. Winther. Gaussian processes for classification: mean-field algorithms. *Neural Computation*, 12(11):2655–2684, November 2000.

[45] Tom Minka and others. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

[46] Xin-She Yang. *Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics*. Cambridge International Science Publishing, 2008.

[47] *Understanding and Using Linear Programming*. Universitext. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[48] Mohammad Taghi Hajiaghayi and Kamal Jain. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 631–640. Society for Industrial and Applied Mathematics, 2006.

[49] Michel X. Goemans and David P. Williamson. Approximation algorithms for np-hard problems. chapter The Primal-dual Method for Approximation Algorithms and Its Application to Network Design Problems, pages 144–191. PWS Publishing Co., Boston, MA, USA, 1997.

[50] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. A fast, adaptive variant of the goemans-williamson scheme for the prize-collecting steiner tree problem. In *Workshop of the 11th DIMACS Implementation Challenge. URL https://people. csail. mit. edu/ludwigs/papers/dimacs14_fastpcst. pdf*, 2014.

[51] Amit Varma and Bernhard O. Palsson. Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. *Nat Biotech*, 12(10):994–998, October 1994.

[52] Kenneth J Kauffman, Purusharth Prakash, and Jeremy S Edwards. Advances in flux balance analysis. *Current Opinion in Biotechnology*, 14(5):491–496, October 2003.

[53] Jeffrey D. Orth, Ines Thiele, and Bernhard Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, March 2010.

[54] Jeremy S. Edwards, Rafael U. Ibarra, and Bernhard Ø. Palsson. In silico predictions of Escherichia coli metabolic capabilities are consistent with experimental data. *Nature Biotechnology*, 19(2):125–130, January 2001.

[55] Patrick F. Suthers, Anthony P. Burgard, Madhukar S. Dasika, Farnaz Nowroozi, Stephen Van Dien, Jay D. Keasling, and Costas D. Maranas. Metabolic flux elucidation for large-scale models using 13c labeled isotopes. *Metabolic Engineering*, 9(5-6):387 – 405, 2007.

[56] Robert L. Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.

[57] Robert L. Smith. The hit-and-run sampler: a globally reaching Markov chain sampler for generating arbitrary multivariate distributions. In *Proceedings of the 28th conference on Winter simulation*, pages 260–264. IEEE Computer Society, 1996.

[58] V. Turchin. On the Computation of Multidimensional Integrals by the Monte-Carlo Method. *Theory Probab. Appl.*, 16(4):720–724, January 1971.

[59] László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM Journal on Computing*, 35(4):985–1005, 2006.

[60] Wout Megchelenbrink, Martijn Huynen, and Elena Marchiori. *optGpSampler*: An improved tool for uniformly sampling the solution-space of genome-scale metabolic networks. *PLoS ONE*, 9(2):e86587, 02 2014.

[61] Alfredo Braunstein and Anna Muntoni. Practical optimization of Steiner trees via the cavity method. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(7):073302, 2016.

[62] Stephan Held, Bernhard Korte, Dieter Rautenbach, and Jens Vygen. Combinatorial optimization in VLSI design. *Combinatorial Optimization-Methods and Applications*, 31:33–96, 2011.

[63] M. Grötschel, A. Martin, and R. Weismantel. The steiner tree packing problem in VLSI design. *Mathematical Programming*, 78(2):265–281.

[64] Matthias W. Seeger. Bayesian inference and optimal design for the sparse linear model. *The Journal of Machine Learning Research*, 9:759–813, 2008.

[65] Andrew S. Kennard, Matteo Osella, Avelino Javer, Jacopo Grilli, Philippe Nghe, Sander J. Tans, Pietro Cicuta, and Marco Cosentino Lagomarsino. Individuality and universality in the growth-division laws of single *E. coli* cells. *Phys. Rev. E*, 93:012408, Jan 2016.

[66] Alfredo Braunstein, Anna Paola Muntoni, and Andrea Pagnani. An analytic approximation of the feasible space of metabolic networks. *Accepted by Nature Communications*, January 2017.

[67] Zachary A King, Justin Lu, Andreas Dräger, Philip Miller, Stephen Federowicz, Joshua A Lerman, Ali Ebrahim, Bernhard O Palsson, and Nathan E Lewis. Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522, 2016.

[68] Nathan E. Lewis, Gunnar Schramm, Aarash Bordbar, Jan Schellenberger, Michael P. Andersen, Jeffrey K. Cheng, Nilam Patel, Alex Yee, Randall A. Lewis, Roland Eils, Rainer König, and Bernhard Ø Palsson. Large-scale in silico modeling of metabolic interactions between cell types in the human brain. *Nature Biotechnology*, 28(12):1279–1285, December 2010.

[69] Natalie C. Duarte, Scott A. Becker, Neema Jamshidi, Ines Thiele, Monica L. Mo, Thuy D. Vo, Rohith Srivas, and Bernhard Ø. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proceedings of the National Academy of Sciences of the United States of America*, 104(6):1777–1782, February 2007.

[70] Jennifer L. Reed, Thuy D. Vo, Christophe H. Schilling, and Bernhard O. Palsson. An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR). *Genome Biology*, 4(9):R54, 2003.

[71] Sharon J. Wiback, Iman Famili, Harvey J. Greenberg, and Bernhard Ø Palsson. Monte Carlo sampling can be used to determine the size and shape of the steady-state flux space. *Journal of Theoretical Biology*, 228(4):437–447, June 2004.

[72] J. Fernandez-de Cossio-Diaz and R. Mulet. Fast inference of ill-posed problems within a convex space. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(7):073207, 2016.

[73] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[74] *Practical Mathematical Optimization*, volume 97 of *Applied Optimization*. Springer-Verlag, New York, 2005.

[75] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[76] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated Anatomical Labeling of Activations in SPM Using a Macroscopic Anatomical Parcellation of the MNI MRI Single-Subject Brain. *NeuroImage*, 15(1):273–289, January 2002.

[77] D. L. Collins, P. Neelin, T. M. Peters, and A. C. Evans. Automatic 3d intersubject registration of MR volumetric data in standardized Talairach space. *Journal of Computer Assisted Tomography*, 18(2):192–205, April 1994.

[78] A. C. Evans, M. Kamber, D. L. Collins, and D. MacDonald. An MRI-Based Probabilistic Atlas of Neuroanatomy. In S. D. Shorvon, D. R. Fish, F. Andermann, G. M. Bydder, and H. Stefan, editors, *Magnetic Resonance Scanning and Epilepsy*, number 264 in NATO ASI Series, pages 263–274. Springer US, 1994. DOI: 10.1007/978-1-4615-2546-2_48.

[79] E. Gouillart, F. Krzakala, M. Mézard, and L. Zdeborová. Belief-propagation reconstruction for discrete tomography. *Inverse Problems*, 29(3):035003, 2013.

[80] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, November 1992.

[81] Adam G. Polak, Janusz Mroczka, and Dariusz Wysoczański. Tomographic image reconstruction via estimation of sparse unidirectional gradients. *Computers in Biology and Medicine*, 81:93–105, February 2017.

[82] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[83] Laura Balzano and Robert Nowak. Blind calibration of sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 79–88. ACM, 2007.

[84] Christophe Schülke, Francesco Caltagirone, and Lenka Zdeborová. Blind sensor calibration using approximate message passing. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(11):P11013, 2015.

[85] L. A. Shepp and B. F. Logan. The Fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, 21(3):21–43, June 1974.

[86] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized Belief Propagation. In *IN NIPS 13*, pages 689–695. MIT Press, 2000.

[87] Manfred Opper and Ole Winther. A Bayesian approach to on-line learning. *On-line Learning in Neural Networks, ed. D. Saad*, pages 363–378, 1998.

[88] Peter S Maybeck. *Stochastic models, estimation, and control*, volume 3. Academic press, 1982.

[89] Steffen L. Lauritzen and Steffen L. Lauritzen. Propagation of Probabilities, Means and Variances in Mixed Graphical Association Models. *Journal of the American Statistical Association*, 87:1098–1108, 1992.